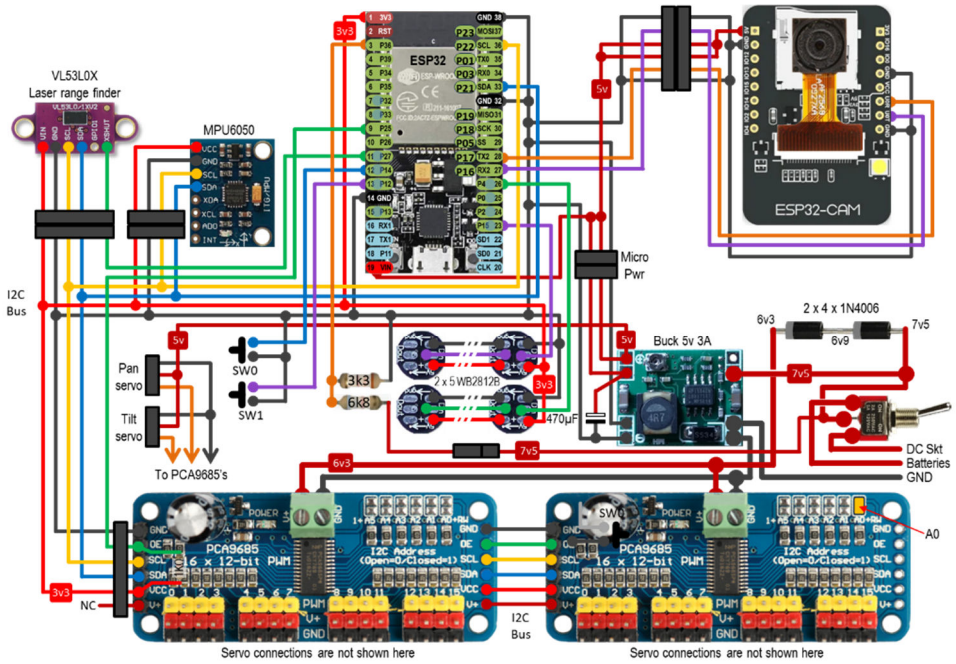
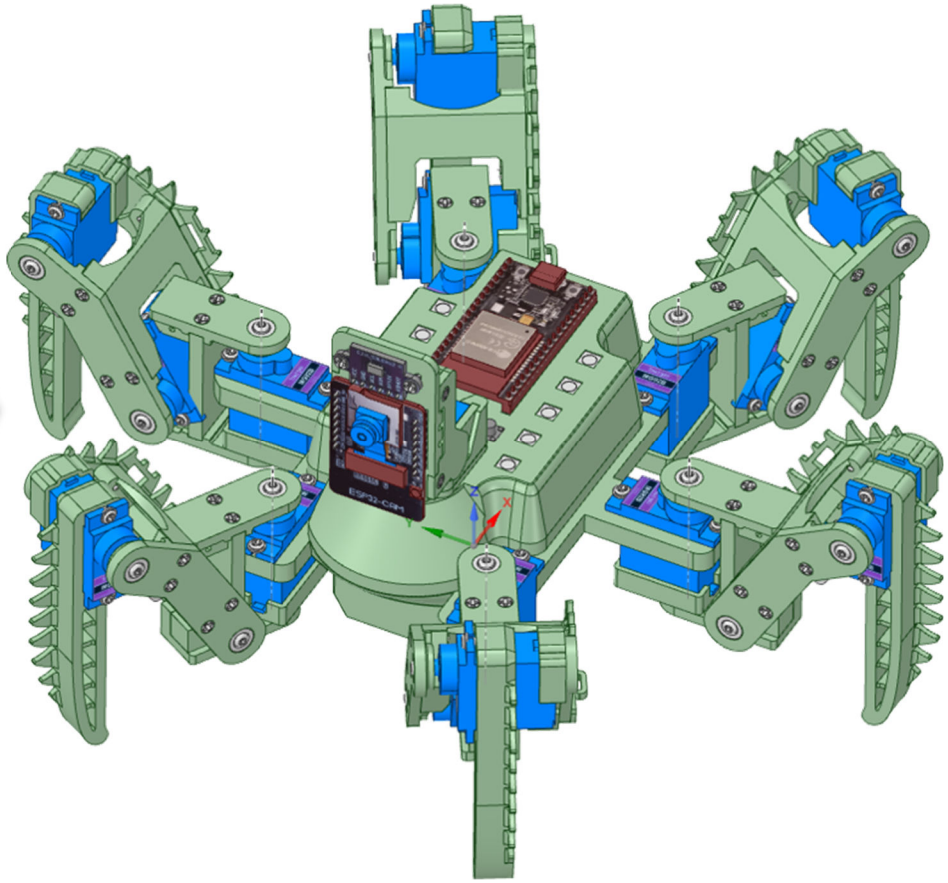
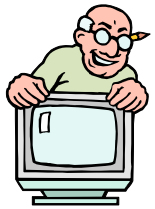
Project SpidaBot

Circuits & Wiring



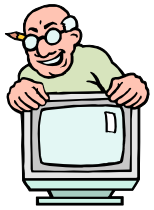




Read through this documentation completely <u>before</u> attempting this project.

Issue: 1.2 Released: 10/05/2025 TechKnowTone

CAUTION



Lithium batteries can be extremely dangerous, if not handled and cared for properly. This design does not include any form of current limiting circuit, like a fuse. So, care must be taken to ensure that the wiring guidelines are followed accurately, that checks are made for short-circuits, and that battery polarities are marked, and they are inserted the correct way round. Failure to do so, could result in an explosive fire.

Charging Practices: Always remove batteries from your project to charge them. Use a charger, designed for the battery used, and from a trusted supplier. Choose a flat, non-flammable surface to charge on, away from flammable materials. Never leave unattended when charging. Don't charge overnight. Monitor charging to ensure charge characteristics are as expected. Only pair batteries with similar characteristics. Do not overcharge, or leave charging for prolonged periods. This increases the risk of damage and fire.

Battery care & maintenance: Stop using a battery if it is swollen, damaged, dented or leaking. Never charge a damaged battery. Never allow a Lithium battery to discharge below 3.2 volts, as cell damage will occur. Avoid extreme temperatures. Do not charge or store batteries in very hot or cold environments. Don't cover batteries whilst charging, as this can trap heat, causing overheating.

In case of fire: Get out and stay out. If a fire starts, leave immediately, and call the fire brigade. For low voltage Lithium batteries, water is a safe extinguisher.

Built-in Monitoring: Most of my project designs include code, and circuitry, to monitor battery voltage, whilst in use. This code then seeks to alert the operator, when the battery has reached a critical low voltage, before shutting down power consuming circuitry; including the micro. Time should therefore be spent on calibrating this feature, as a precaution, for good battery management and maintenance.

Carefully dispose of batteries that have been discharged below their critical voltage.

Hand Tools:

Recommended:

- Fine nosed pliers
- Side cutters
- 1.5 mm Drill
- 2.0 mm Drill
- 4.0 mm Drill
- Needle files
- Screwdrivers
- Craft knife

Note: Not all items needed are shown here.

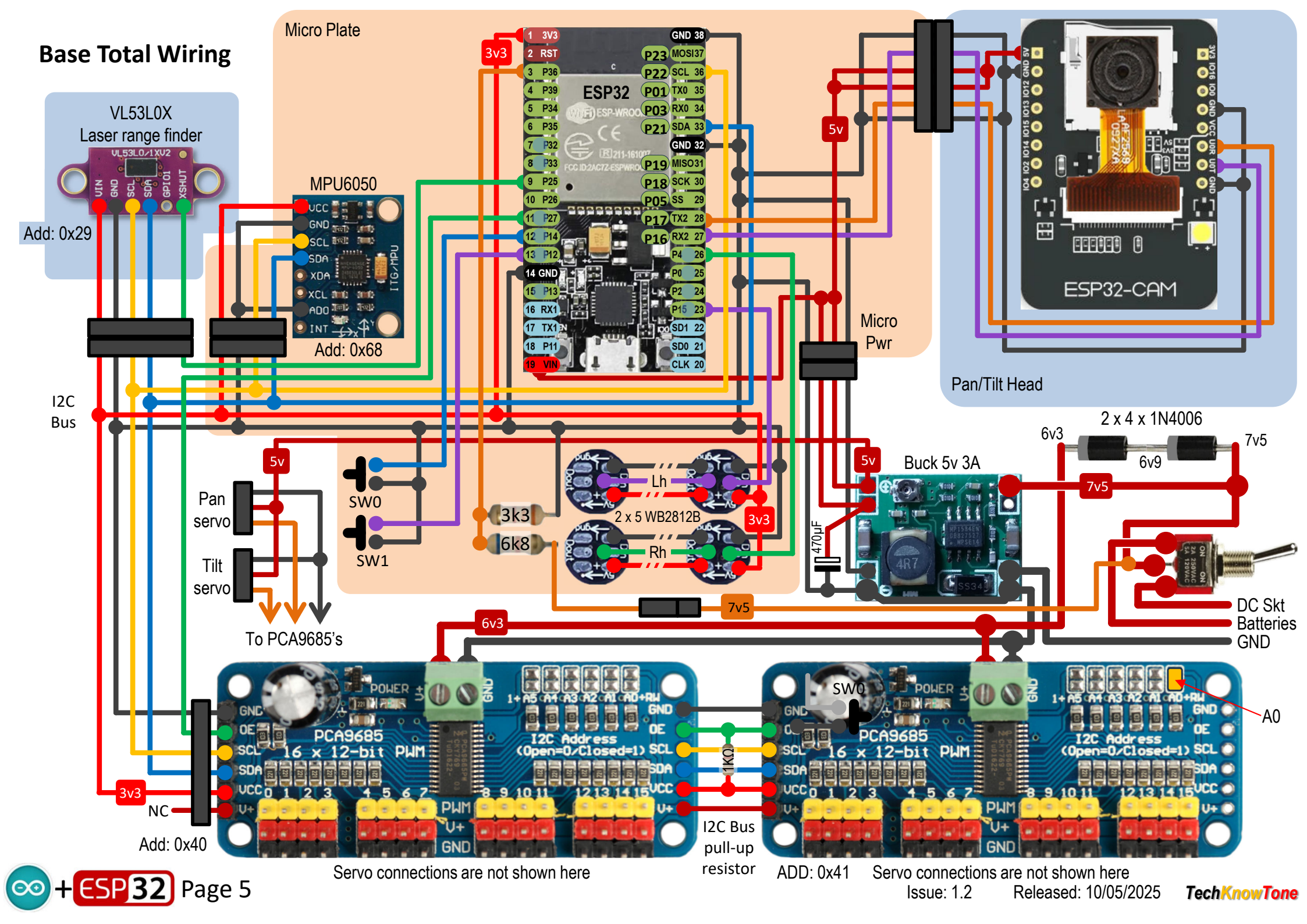
Some printed components act as aids and gauges. Use them.

Tools & Materials:

Temperature controlled iron
Solder flux
Resin cored solder
Hot melt glue gun {optional)
2-part epoxy resin glue
Screw drivers
Wire wrapping tool
Wire wrapping wire 30 AWG
24 AWG stranded wire (red & black)
Multimeter

Issue: 1.2 Released: 10/05/2025

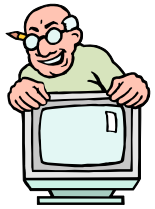
Base Total Wiring



Issue: 1.2 Released: 10/05/2025

TechKnowTone

PCA9685 power lines



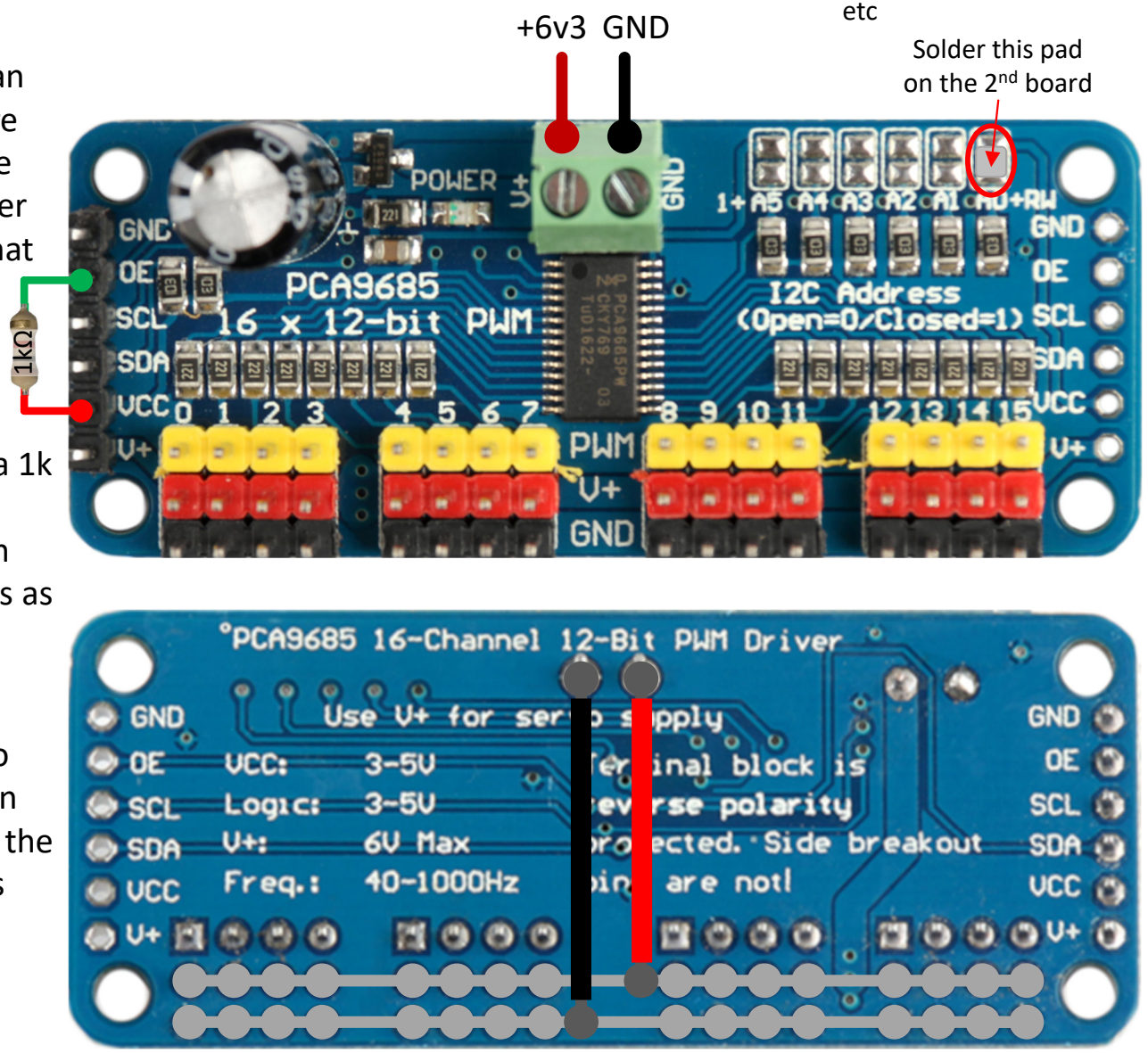
The assignment of servos to PWM channels has been done symmetrically, in such a way as to ease the cable runs to the servos in each leg. See a later slide that details this. As 20 servos are needed in this project, and only 16 channels are available in one PCA9685 board, we need to use two of them in SpidaBot.

Default I2C address = 0x40
with A0 shorted = 0x41
with A1 shorted = 0x42
with A2 shorted = 0x44
etc

The interface to this board is an I2C bus. As more than 16 channels are needed, the two PCA9685 boards are connected in series to extend the I2C bus, and on the 2nd board the A0 address bit is set to '1', using a solder blob, which changes the address on the I2C bus of that board from 0x40 to 0x41, with the LSB set.

The output enable OE is active LOW, and pulled LOW by a 10k resistor on each board. To prevent servo jump at powerup, we pull this pin HIGH using a 1k resistor, as the output pins of the ESP32 will be high impedance until they are defined in code. The OE pin can then be easily driven LOW, to enable both boards as needed in your code.

All of the signals entering the left-hand side of the board, are carried through to the right-hand edge, so that several boards can be used in cascade, with even more servos. I decided to mount the 1kΩ resistor on the boards inter-connecting harness, as seen later in this document.

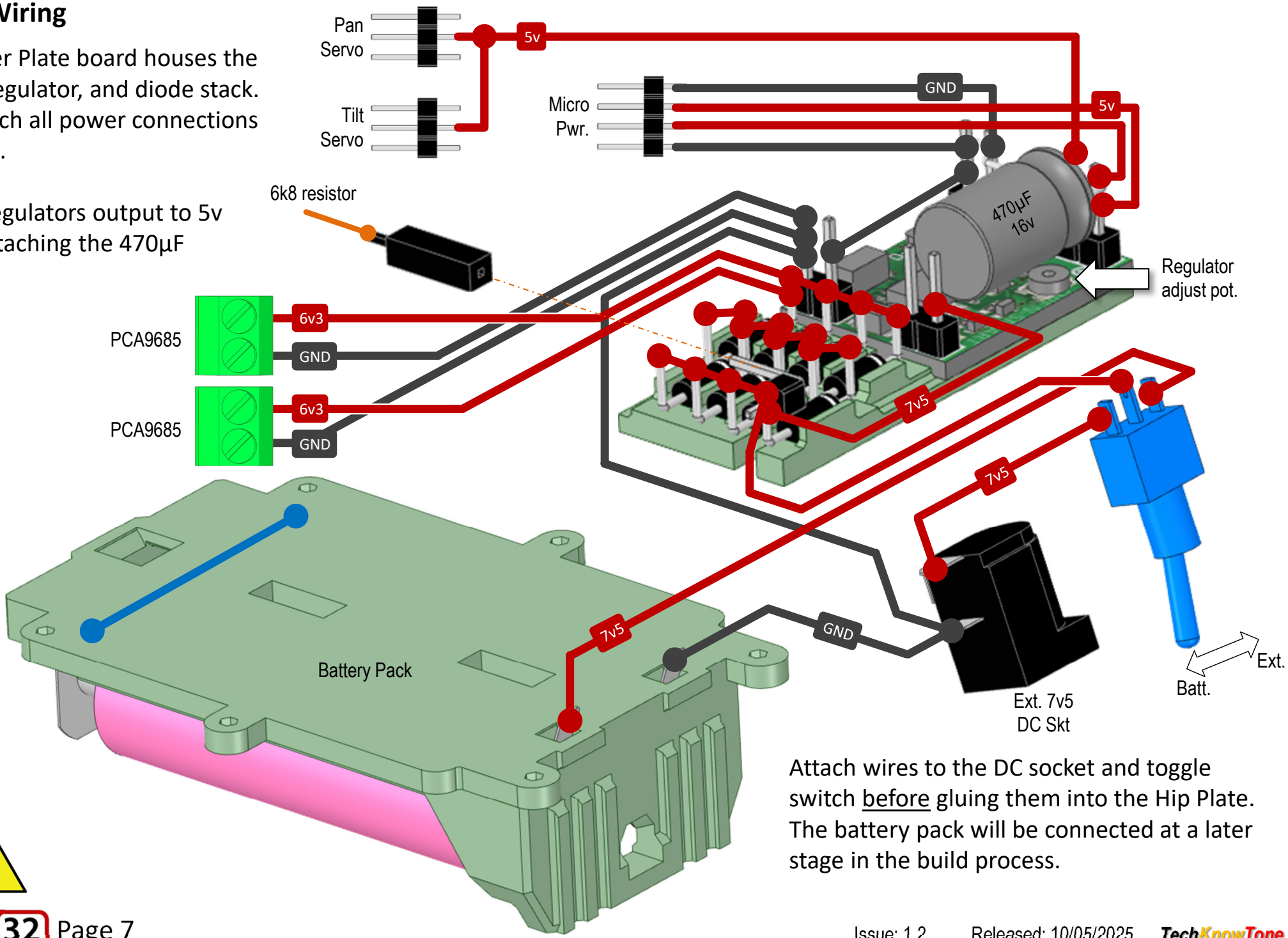


Issue: 1.2 Released: 10/05/2025

Power Wiring

The Power Plate board houses the voltage regulator, and diode stack. From which all power connections are made.

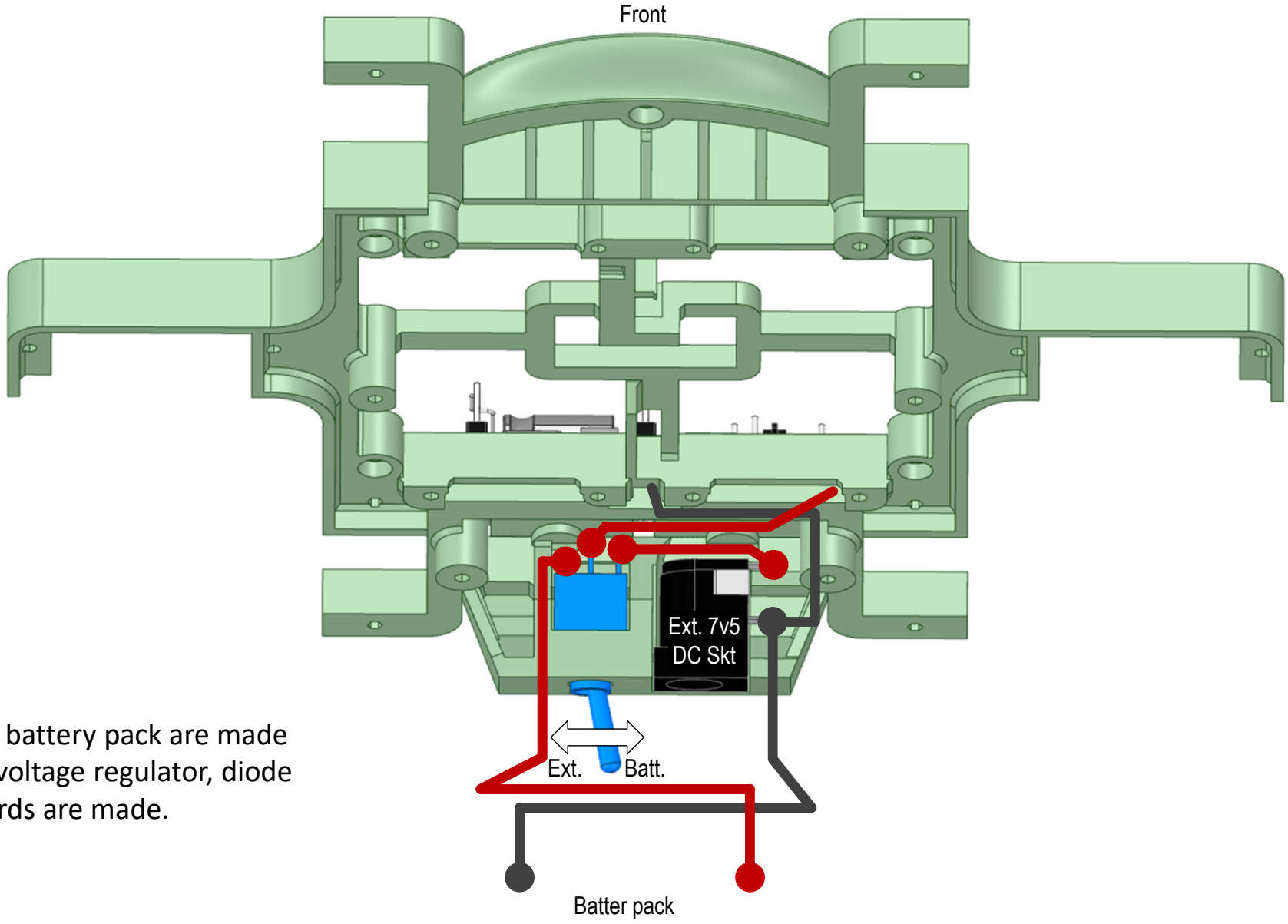
Set the regulators output to 5v before attaching the 470µF capacitor.



Attach wires to the DC socket and toggle switch before gluing them into the Hip Plate. The battery pack will be connected at a later stage in the build process.

Power Wiring – underside view

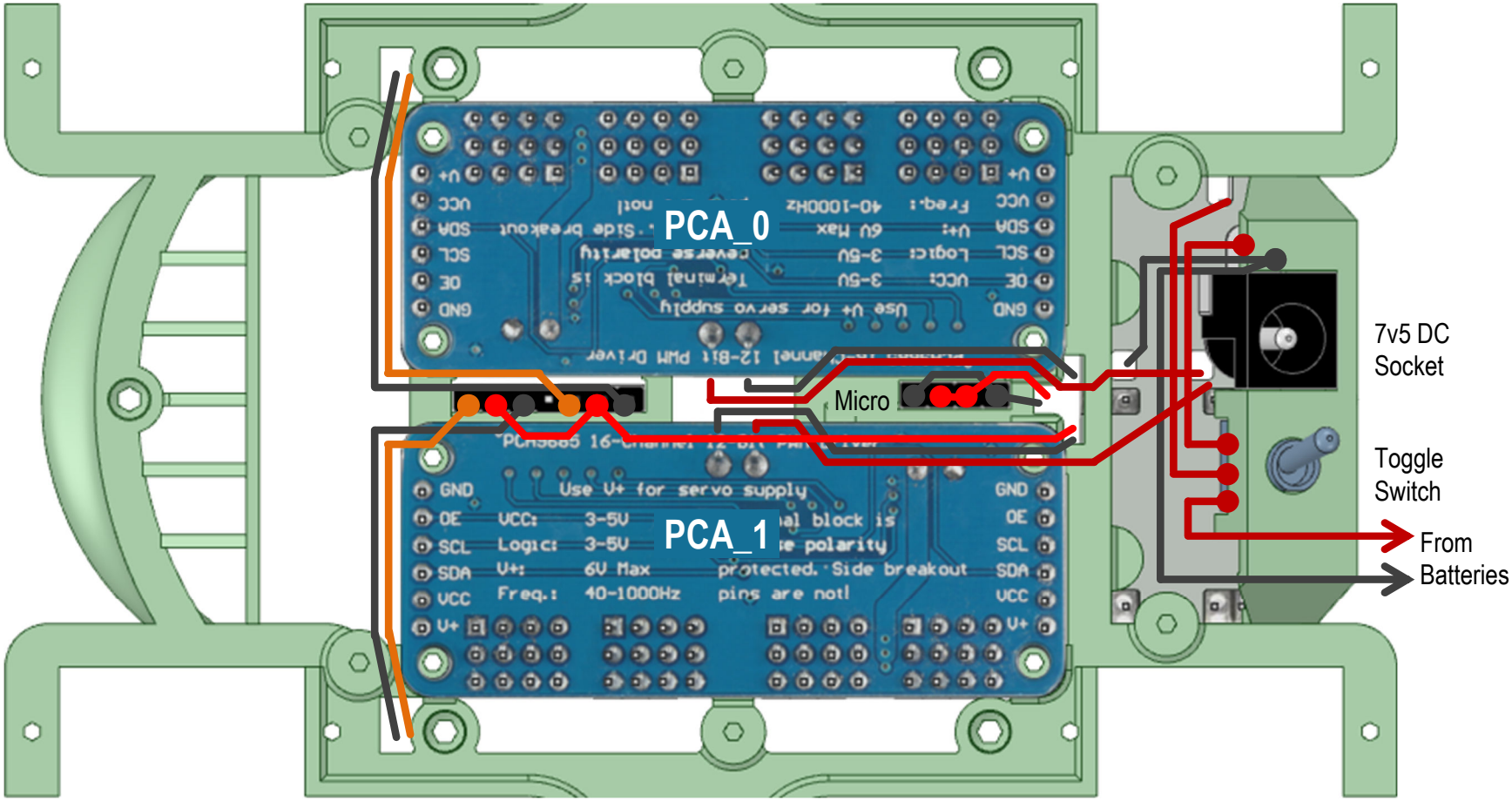
To help your understanding, this is the same information, but view from a different angle.



The connections to the battery pack are made once the wiring to the voltage regulator, diode pack and PCA9685 boards are made.

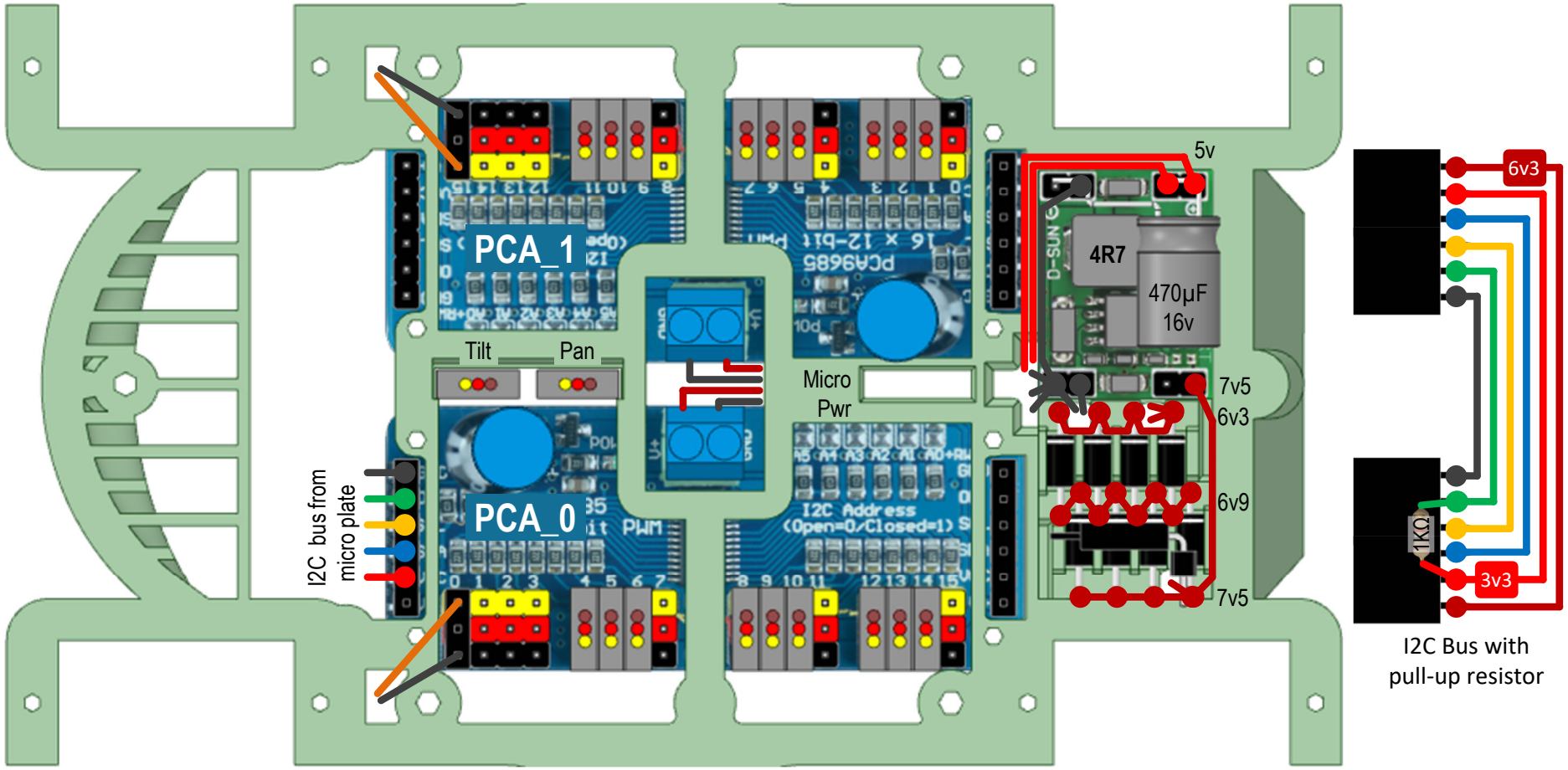
Hip Plate Wiring – underside view

The power is routed from the regulator board as shown. It feeds the 5vDC to the 4-pin strip for the upper body, and 6v3 DC to the two terminal blocks on the PCA9685 boards. A 5v DC feed is also routed to the two 3-pin strips, for the pan and tilt servos. Note that their GND and control signals come from plugs on the other side of the boards.



Hip Plate Wiring – Top View

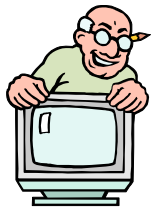
Is shows the chassis plate viewed from above, with the incoming I2C connections on the lower left, fed from the upper micro plate. The wiring of the diode stack and the regulator are shown, along with the short harness that connects the two boards together. Note the mounting of the 1kΩ resistor on this harness.

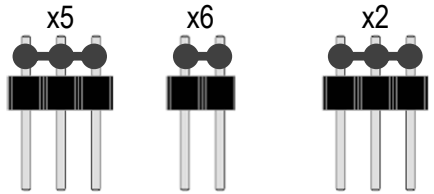


I2C Bus with
pull-up resistor

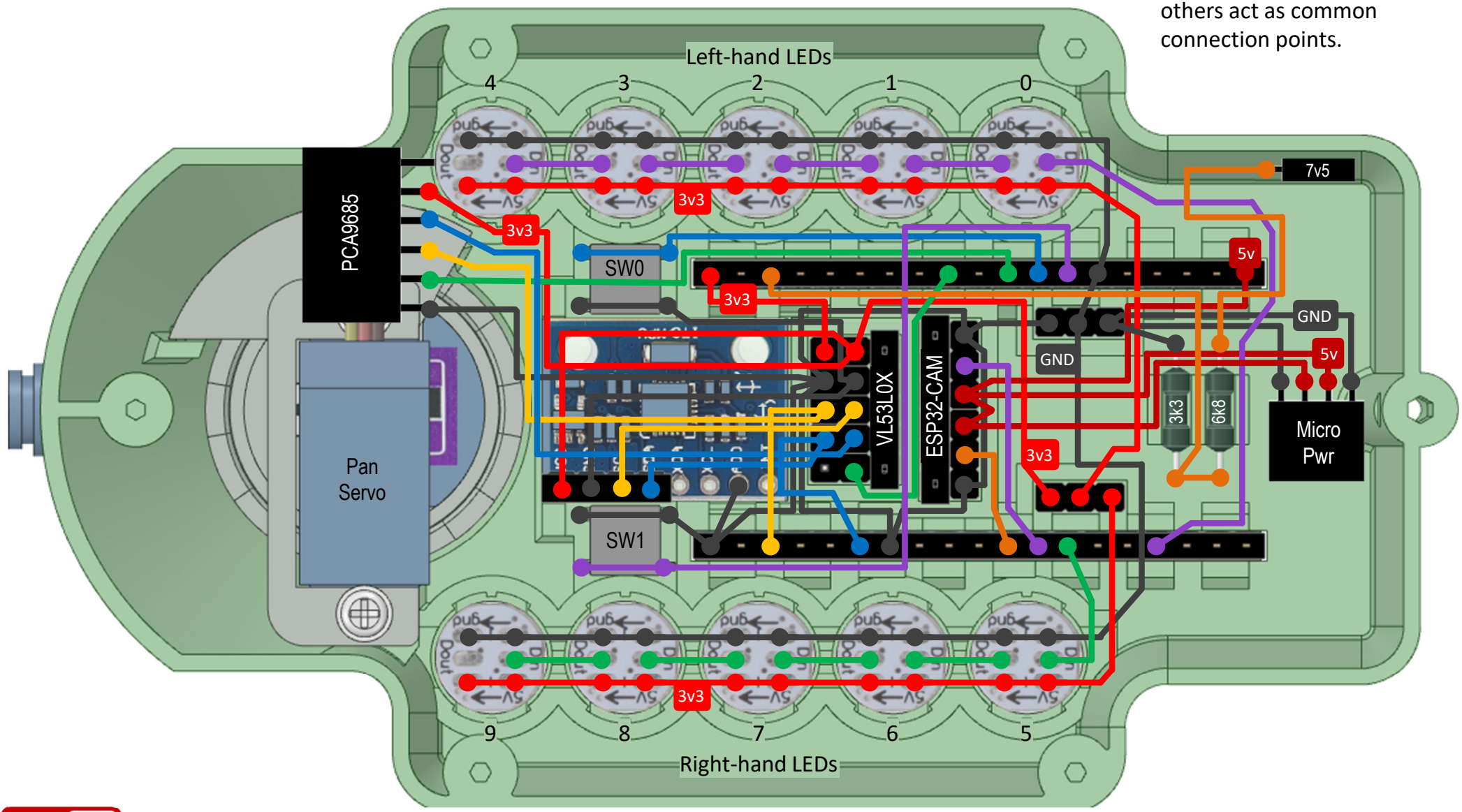
Micro Plate Total Wiring – underside view

The micro plate houses most of the electronics circuitry, taking its 5v power supply from the Hip plate using a 4 pin socket. The ESP32 regulator then provides 3v3 supply for things like the MPU6050, and RGB LEDs.



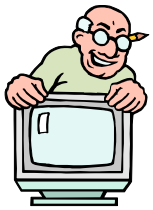


Pin strips are pre-wired, shorting the pins together, before being glued into the micro plate. Some pin strips are then arranged to act as a bus for the I2C devices, whilst others act as common connection points.

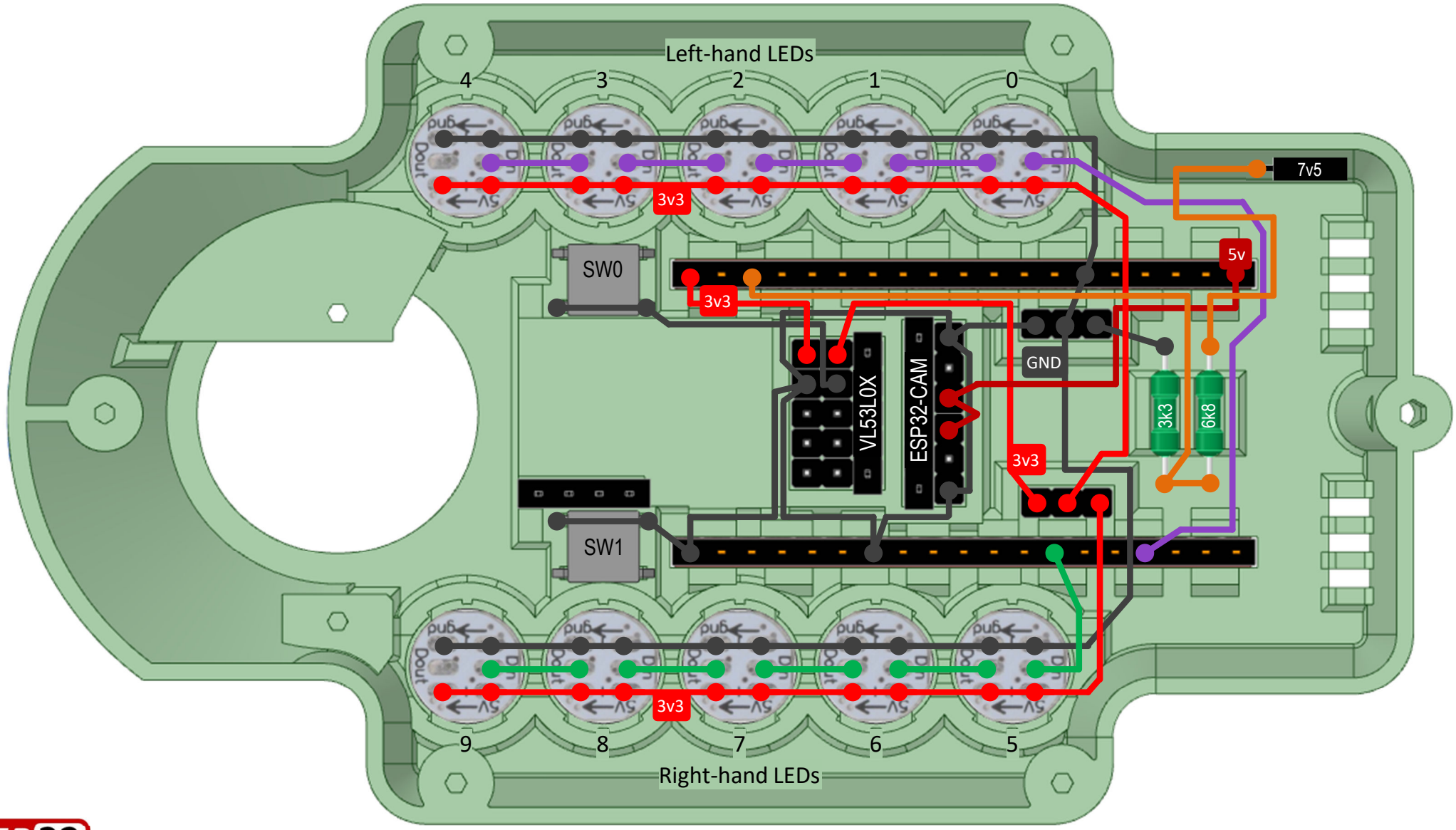


Issue: 1.2 Released: 10/05/2025

Micro Plate Total Wiring – Stage 1



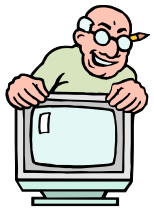
The first stage of the build process is to run in the power connections, and to terminate the LED strips, which were pre-wired in the jig. Whilst 30 AWG wire is used for most of the wiring, the power connections are made with thicker wires.

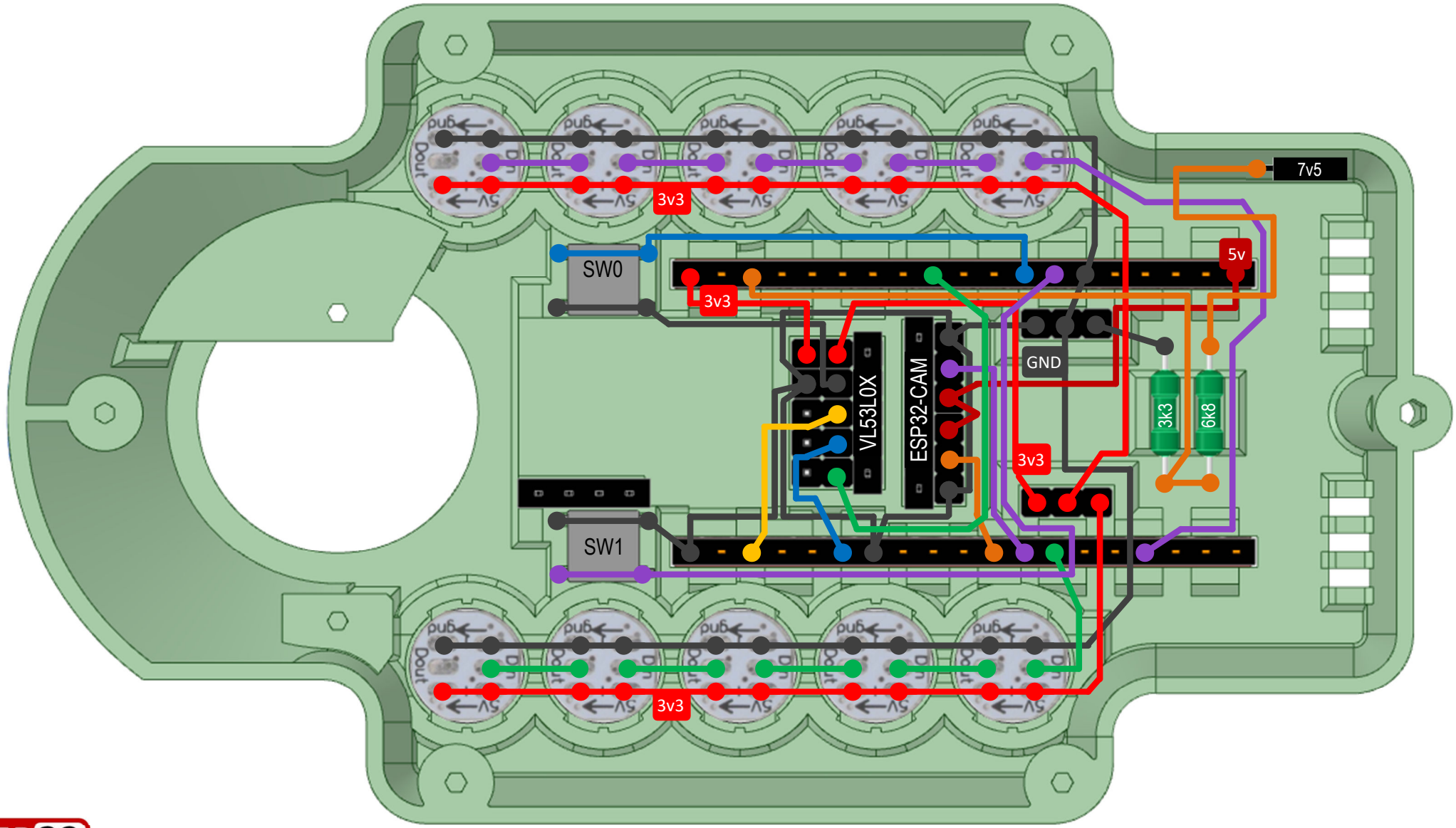


Issue: 1.2 Released: 10/05/2025

Micro Plate Total Wiring – Stage 2

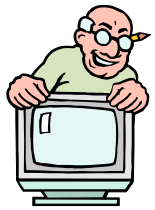
The second stage of the build process is to wire in the I2C bus connections, and those being fed to the ESP32-CAM.



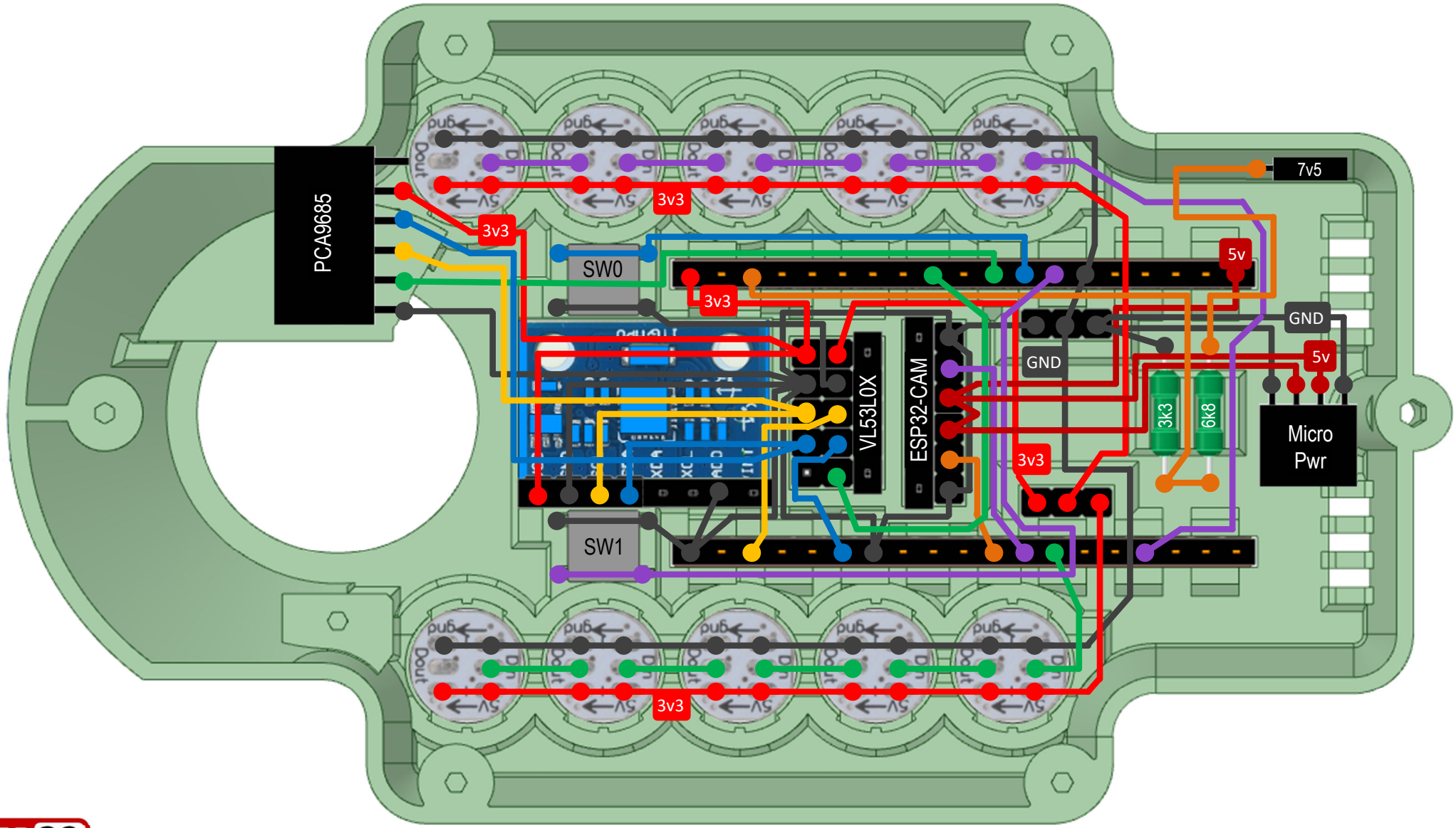


Issue: 1.2 Released: 10/05/2025

Micro Plate Total Wiring – Stage 3



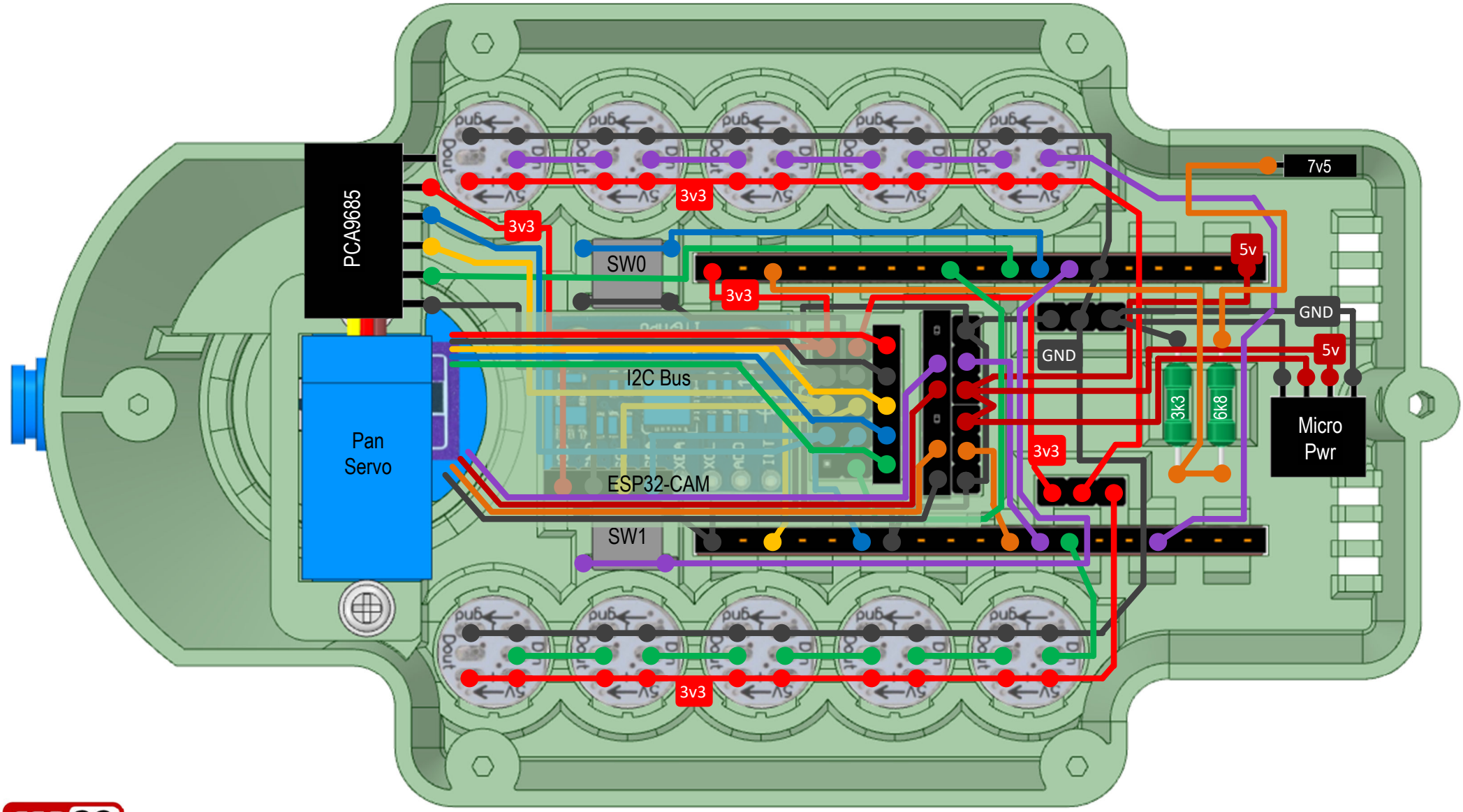
The third stage of the build process is to glue in and wire the MPU6050 motion sensor, and the pin-strip plugs for the PCA9685 I2C bus, and the power to the micro plate. Note that the AD0 line on the MPU6050 must be connected to GND.



Issue: 1.2 Released: 10/05/2025 TechKnowTone

Micro Plate Total Wiring – Stage 4

The forth stage of the build process is to attach the head to the micro plate, using the retaining ring, which is glued on. Then wiring in the VL53L0X range finder, and the ESP-CAM, through the rotary junction to the two pin strip sockets. Previous wiring is faded here to make things clearer.

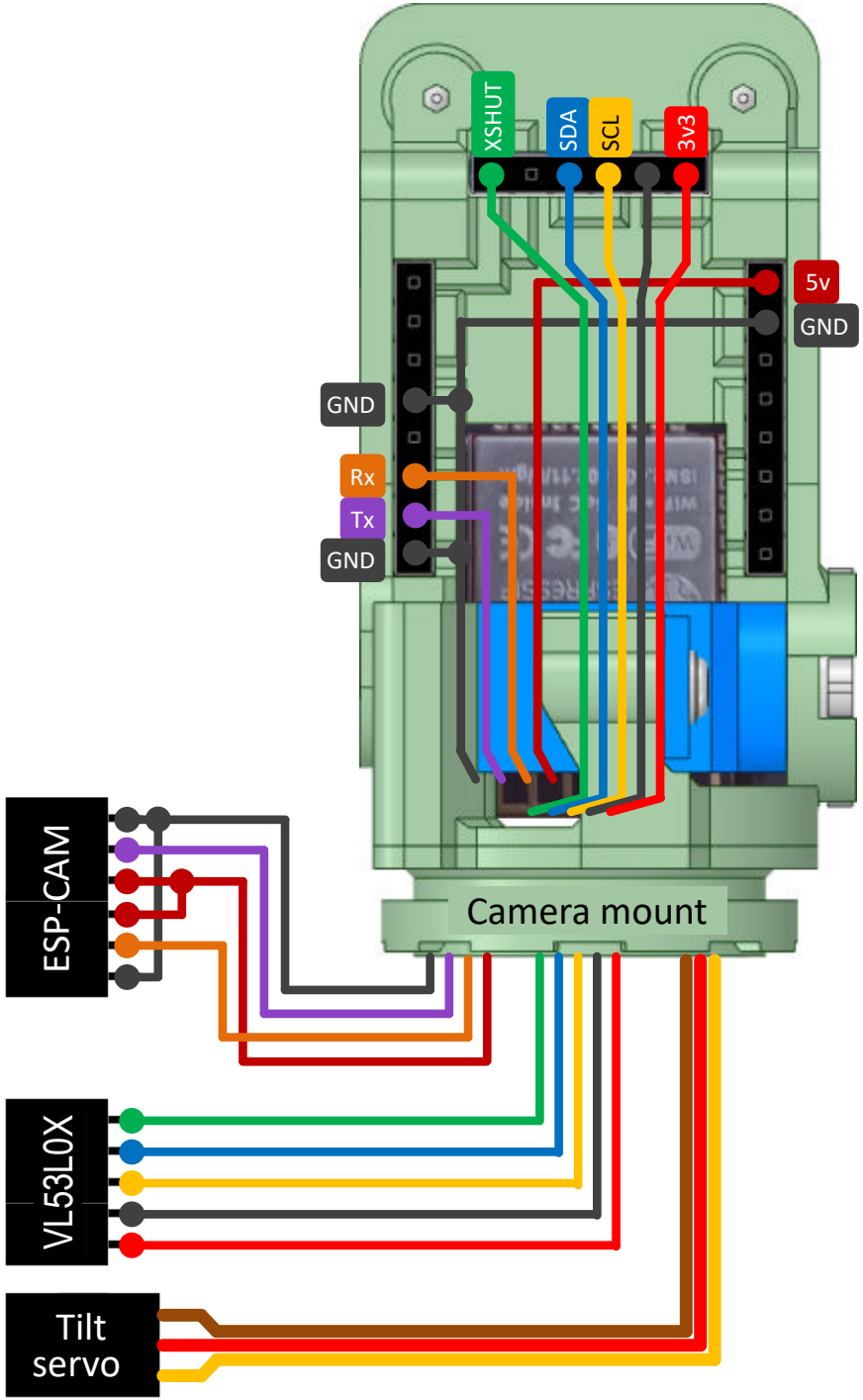


Head Wiring

With the camera mount retaining ring glued into position, the wiring of the ESP-CAM and VL53L0X sensor can commence. We use 30 AWG wire wrap wire for this, as there is not a lot of space though the rotary junction.

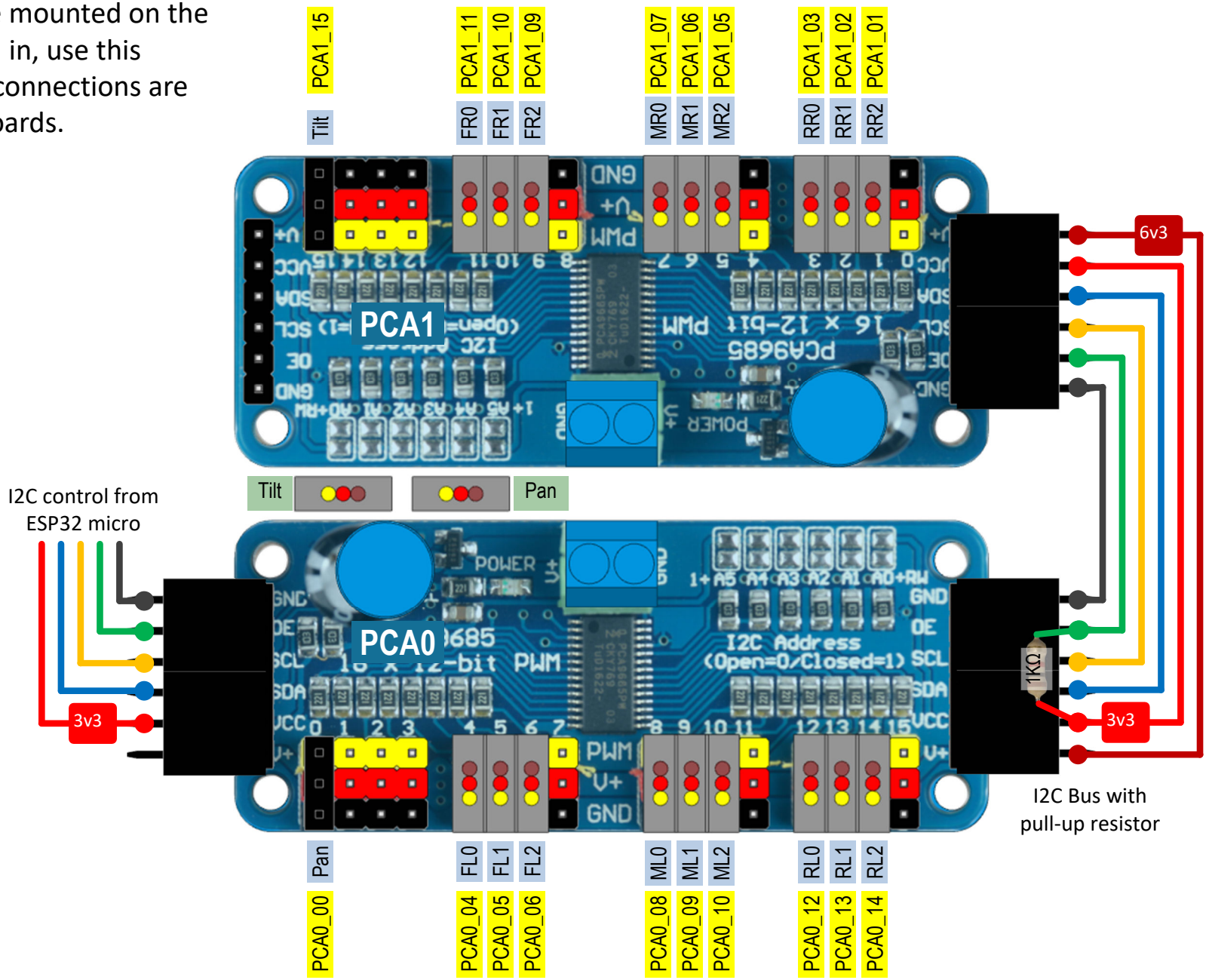
The ESP-CAM wiring is terminated in a 6-pin socket, whilst the VL53L0X sensor is terminated in a 5-pin socket. These then plug into their corresponding pin strips, in the micro plate.

You can check that everything is working, by running the code, before soldering the wire-wrap connections.



Servo Channel Assignments

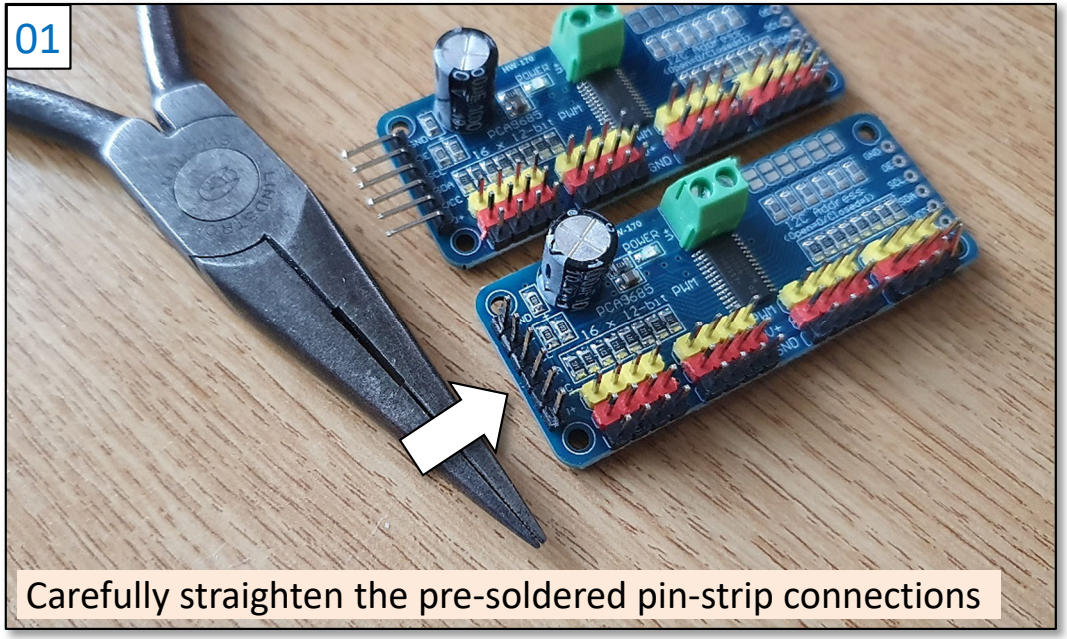
When the MG92B servos are mounted on the legs and ready to be plugged in, use this diagram to show you which connections are used on the two PCA9685 boards.



Issue: 1.2 Released: 10/05/2025 TechKnowTone

Build Images

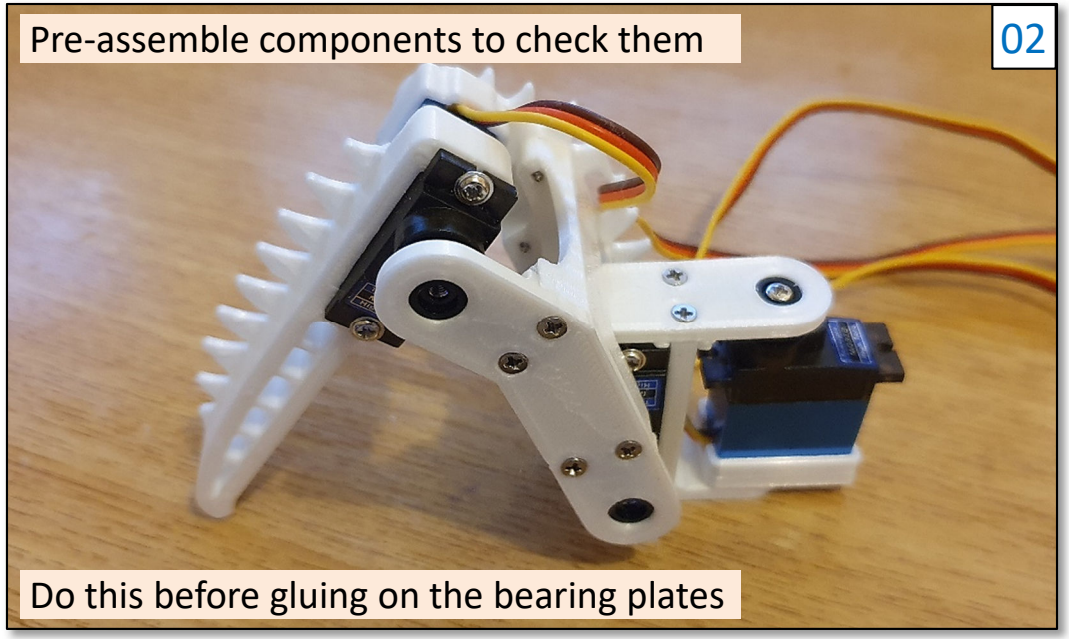
01



Carefully straighten the pre-soldered pin-strip connections

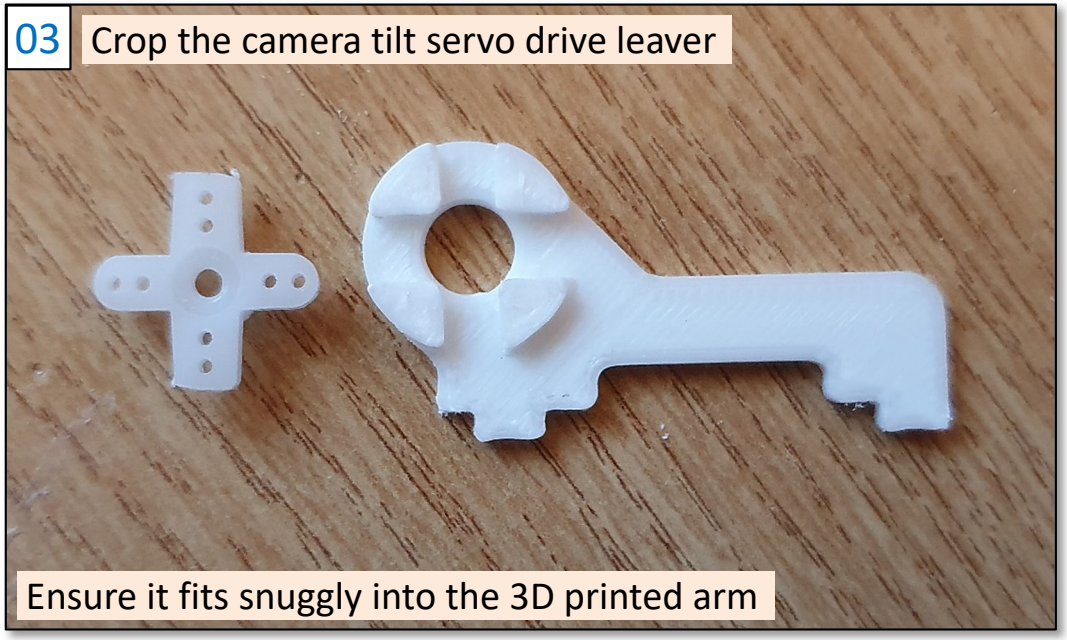
Pre-assemble components to check them

02



Do this before gluing on the bearing plates

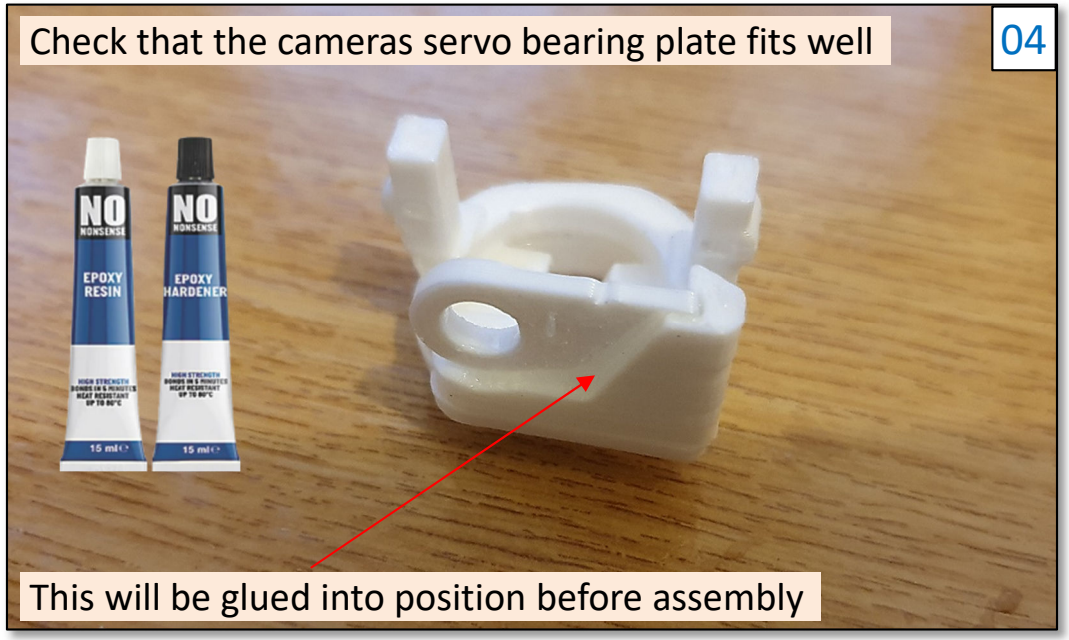
03 Crop the camera tilt servo drive leaver



Ensure it fits snuggly into the 3D printed arm

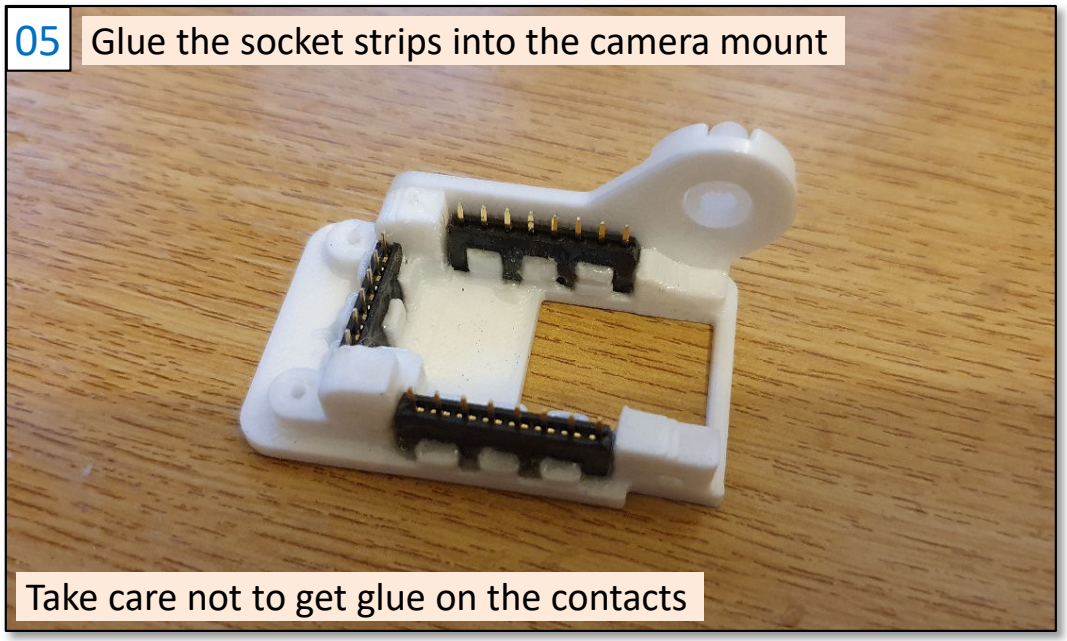
Check that the cameras servo bearing plate fits well

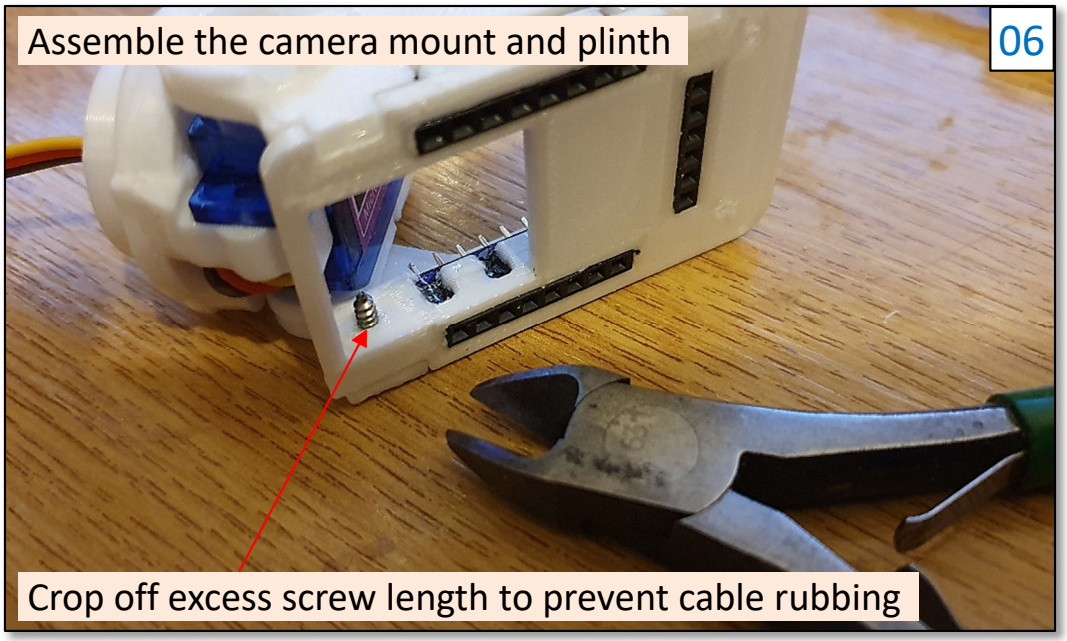
04

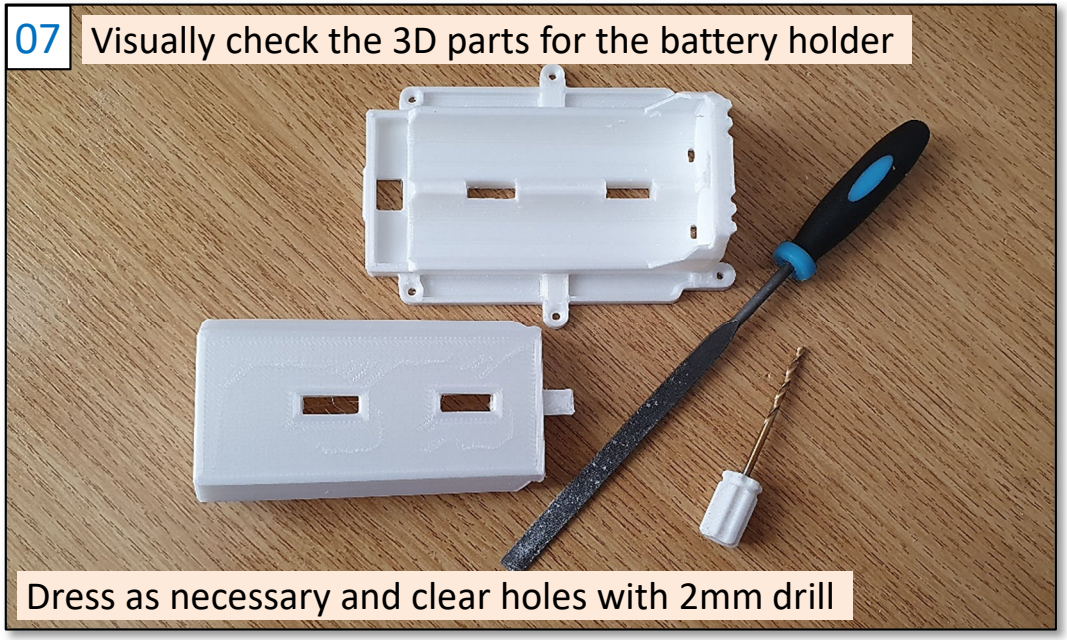


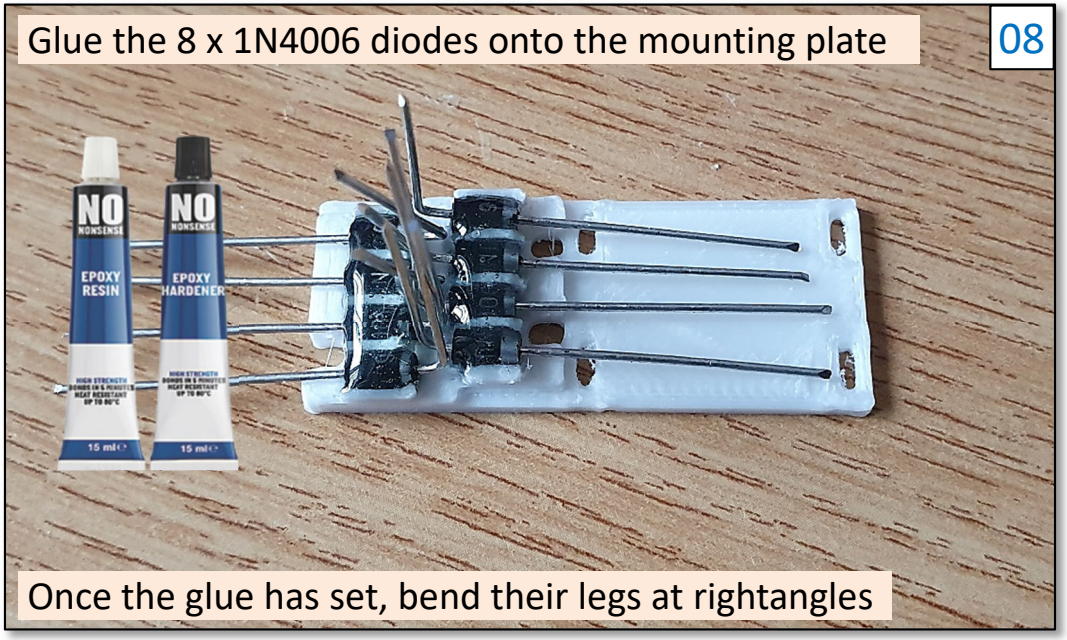
This will be glued into position before assembly

Build Images



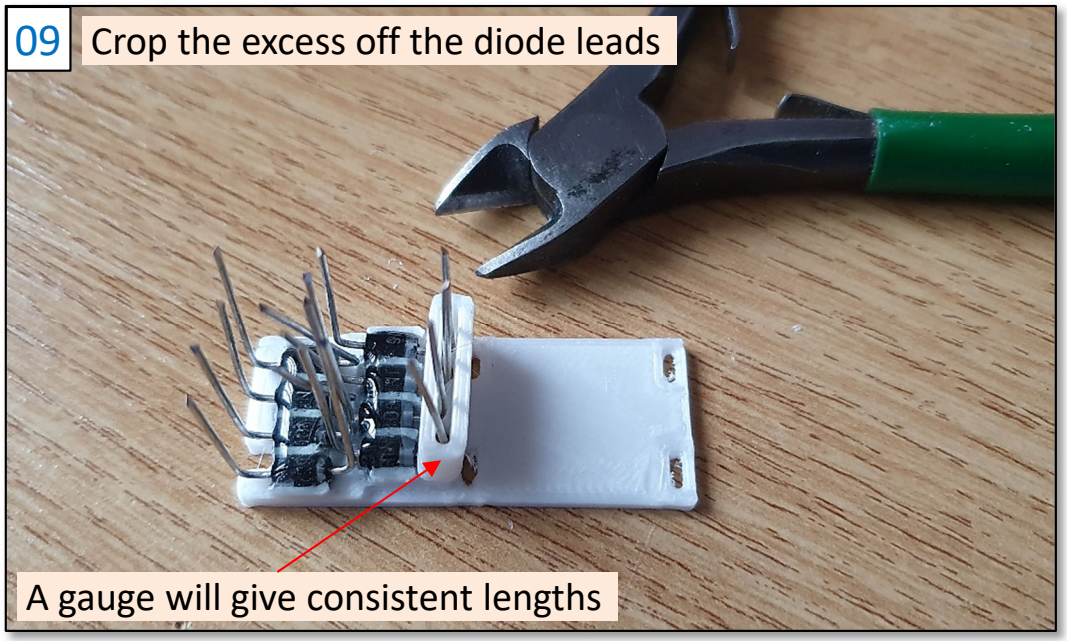






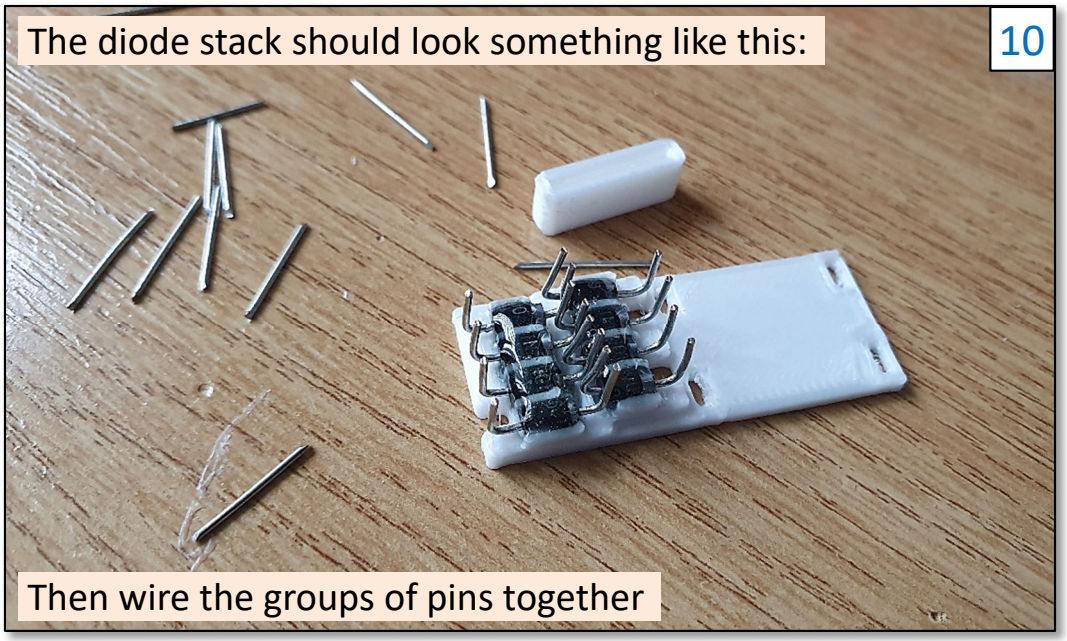
Build Images

09 Crop the excess off the diode leads



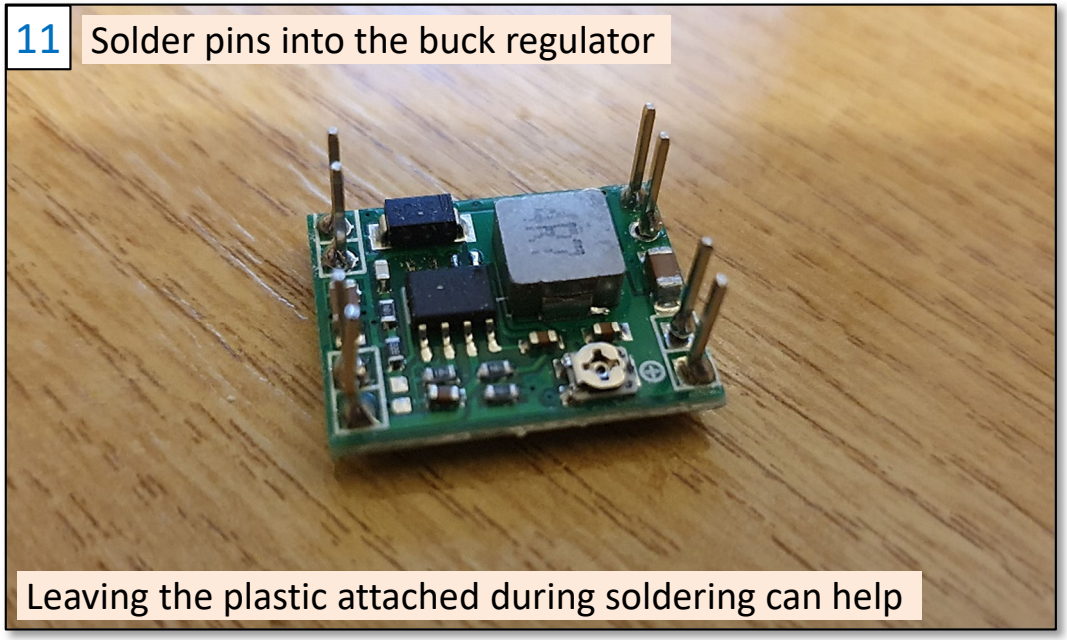
A gauge will give consistent lengths

The diode stack should look something like this: 10



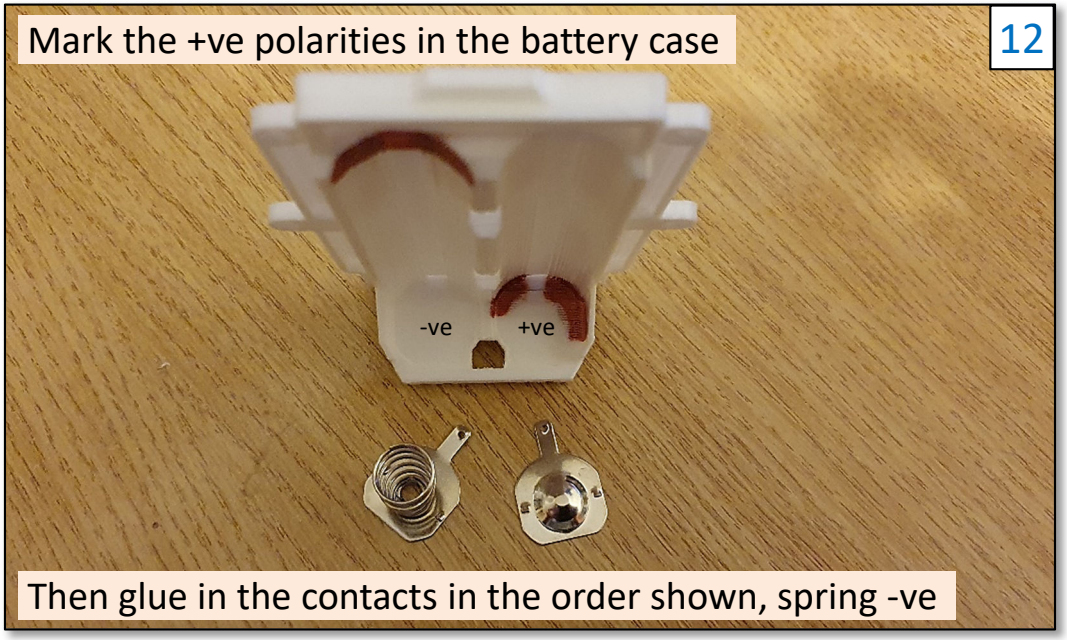
Then wire the groups of pins together

11 Solder pins into the buck regulator



Leaving the plastic attached during soldering can help

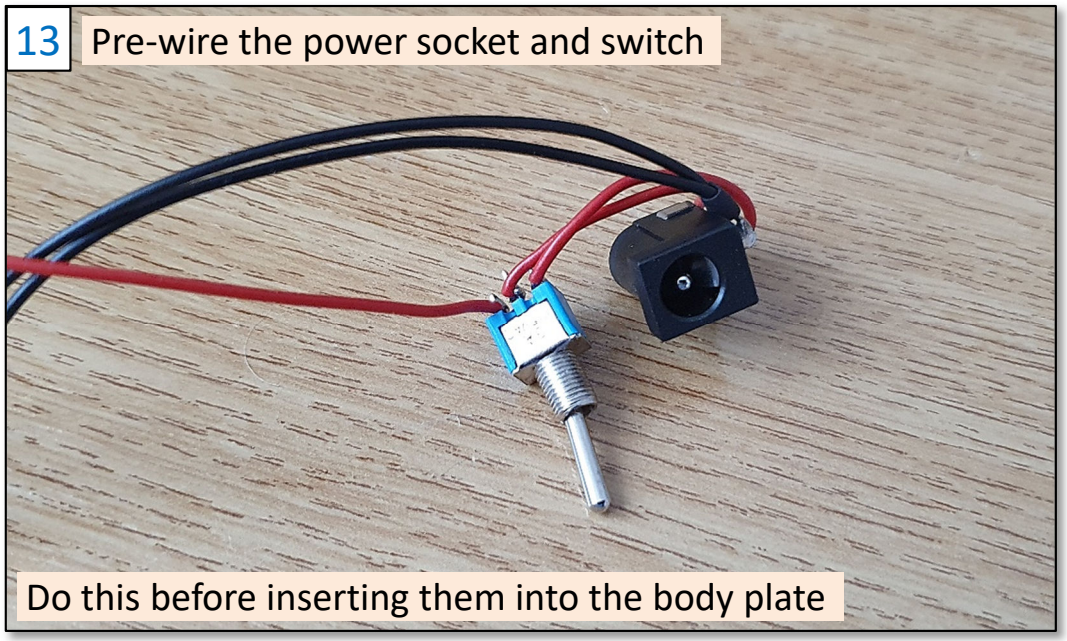
Mark the +ve polarities in the battery case 12



Then glue in the contacts in the order shown, spring -ve

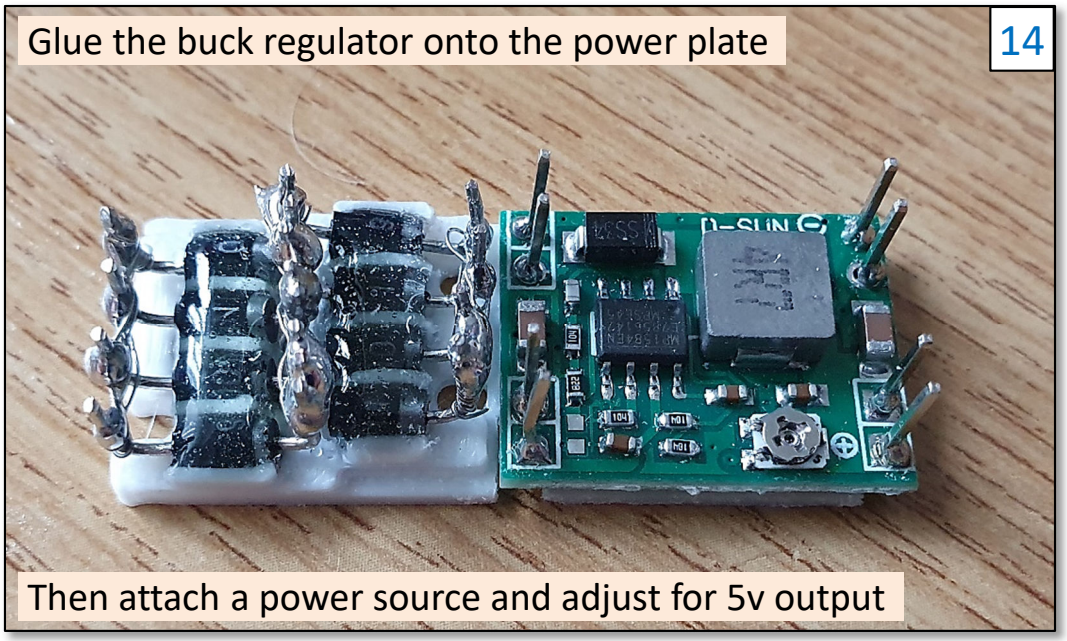
Build Images

13 Pre-wire the power socket and switch



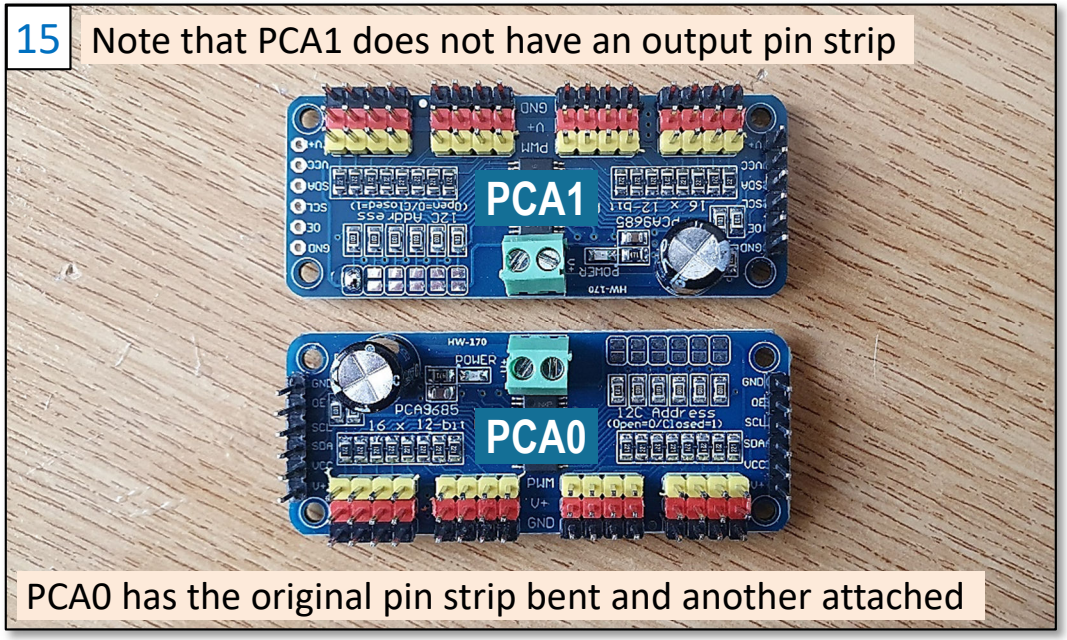
Do this before inserting them into the body plate

Glue the buck regulator onto the power plate 14



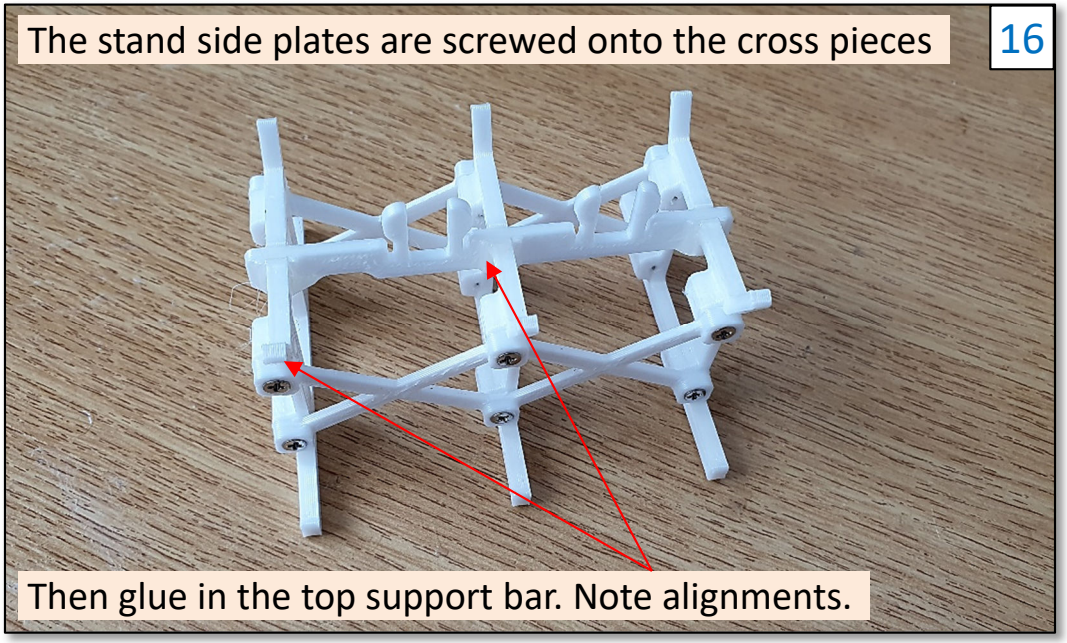
Then attach a power source and adjust for 5v output

15 Note that PCA1 does not have an output pin strip



PCA0 has the original pin strip bent and another attached

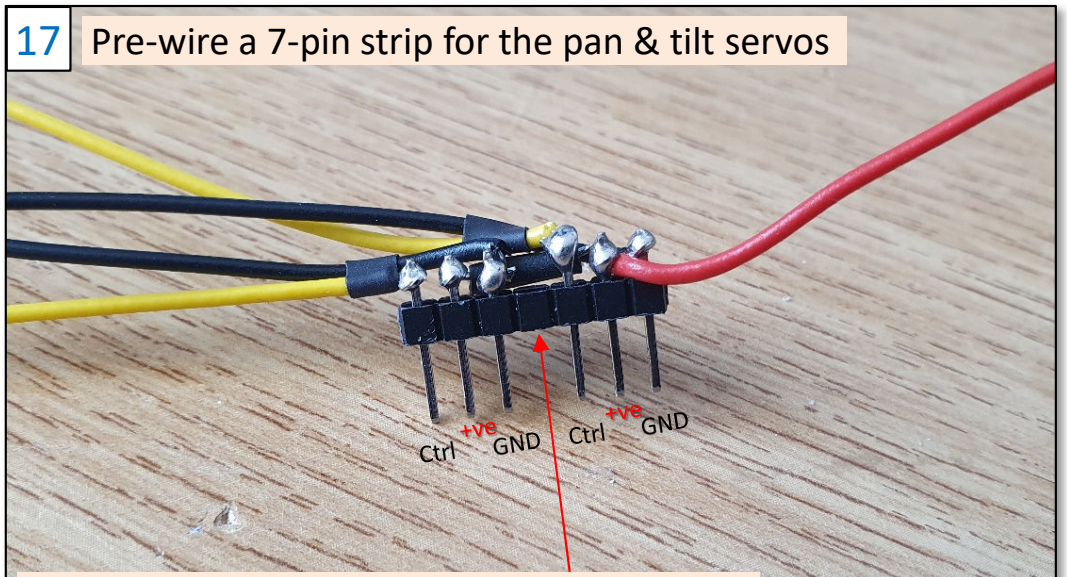
The stand side plates are screwed onto the cross pieces 16



Then glue in the top support bar. Note alignments.

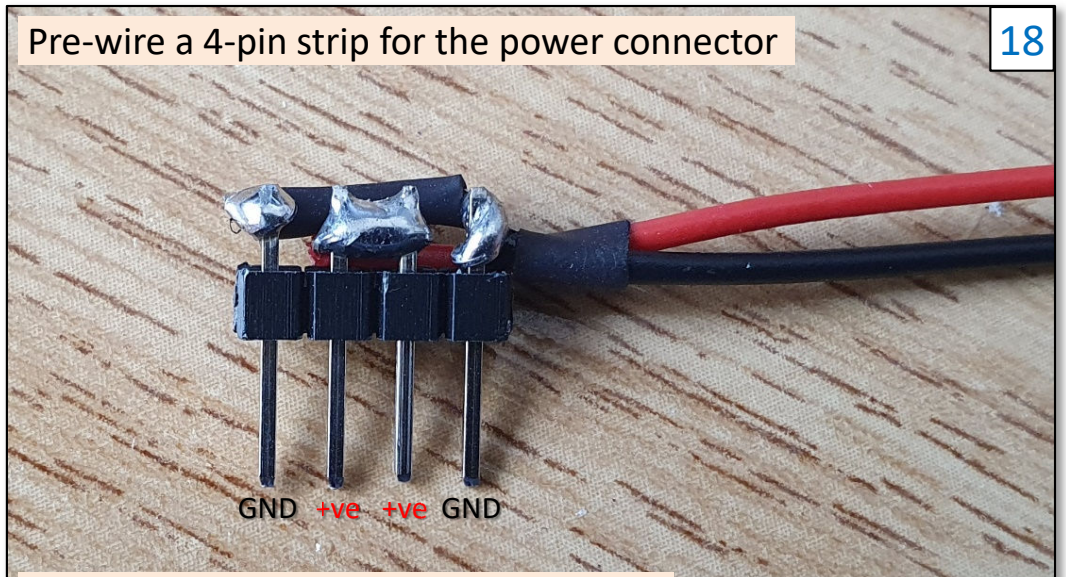
Build Images

17 Pre-wire a 7-pin strip for the pan & tilt servos



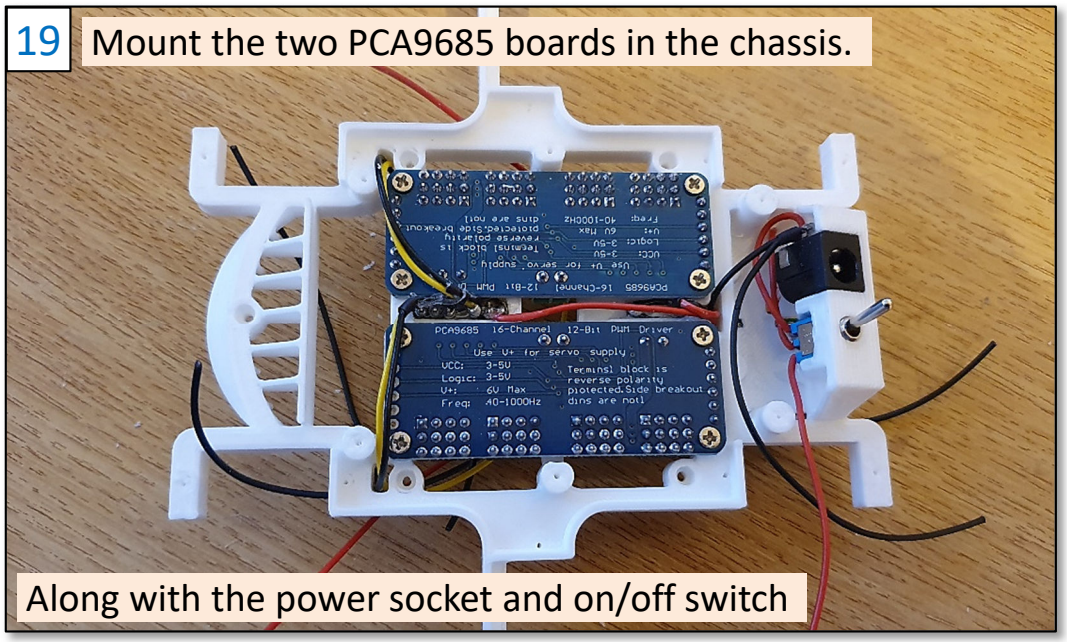
Note that the centre pin has been removed.

Pre-wire a 4-pin strip for the power connector 18



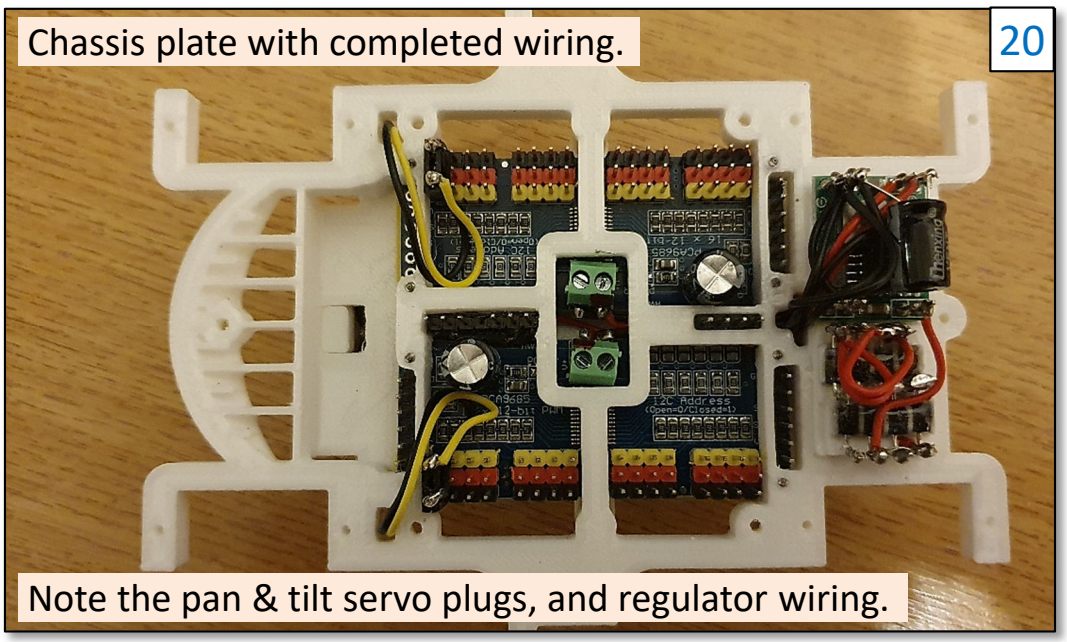
This is later glued into the base chassis

19 Mount the two PCA9685 boards in the chassis.



Along with the power socket and on/off switch

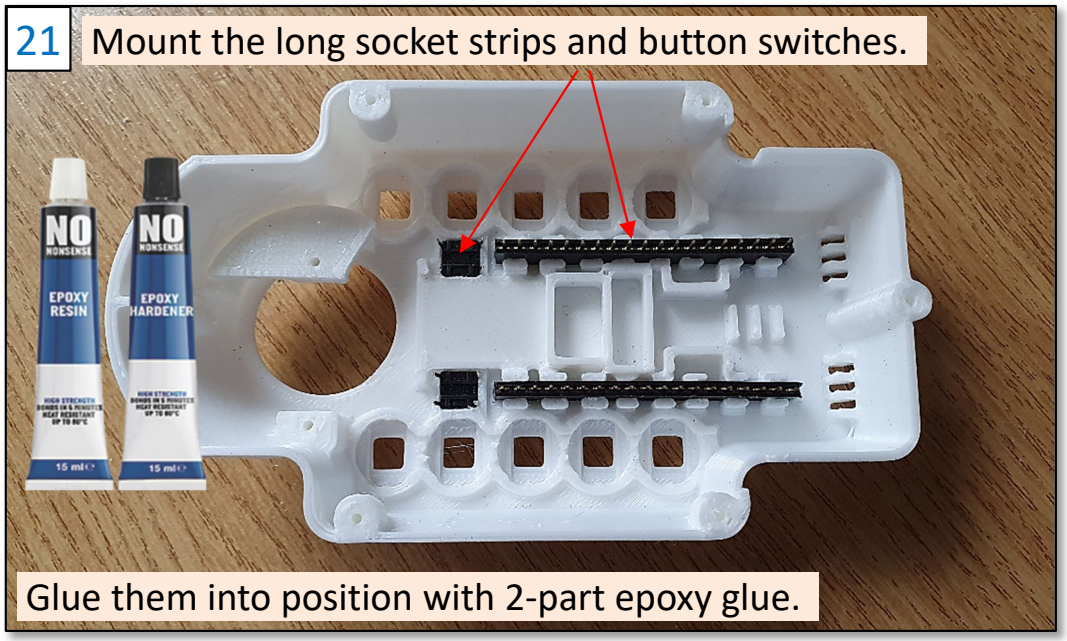
Chassis plate with completed wiring. 20



Note the pan & tilt servo plugs, and regulator wiring.

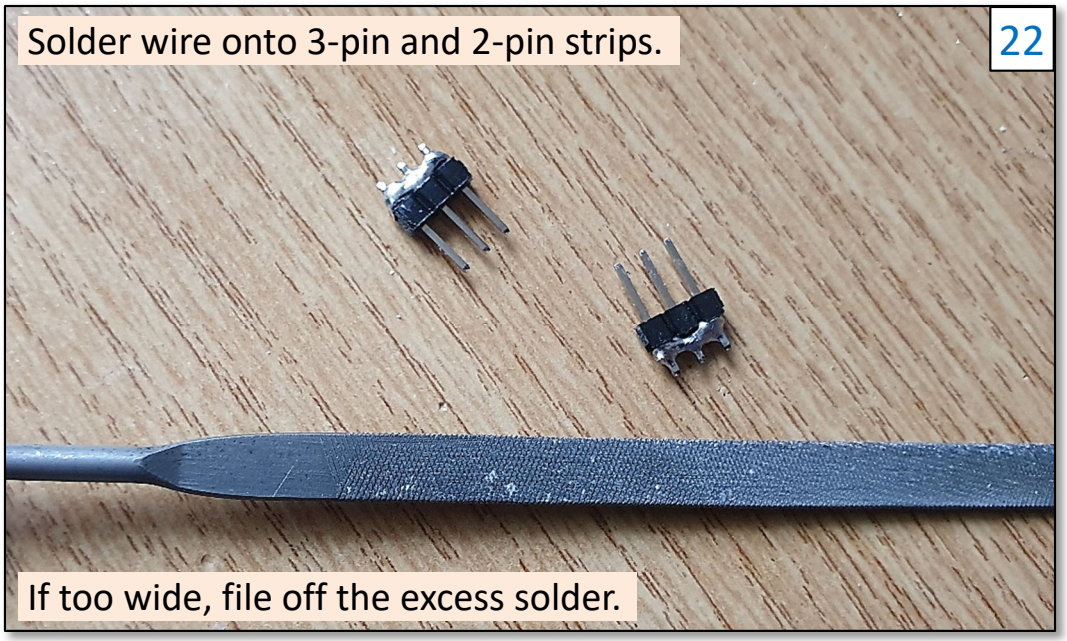
Build Images

21 Mount the long socket strips and button switches.



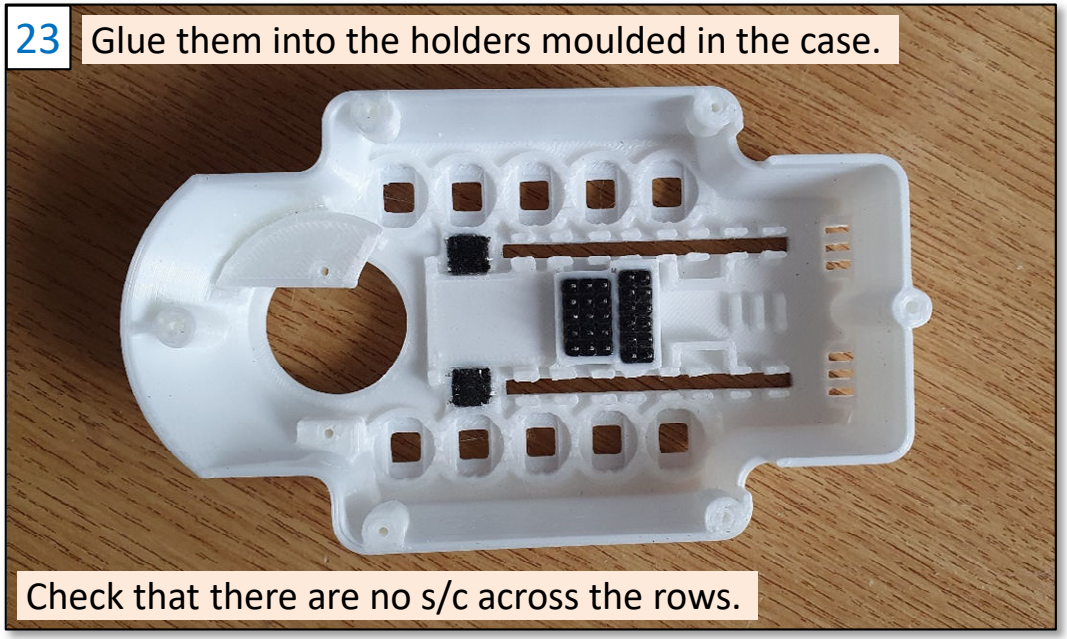
Glue them into position with 2-part epoxy glue.

Solder wire onto 3-pin and 2-pin strips. 22



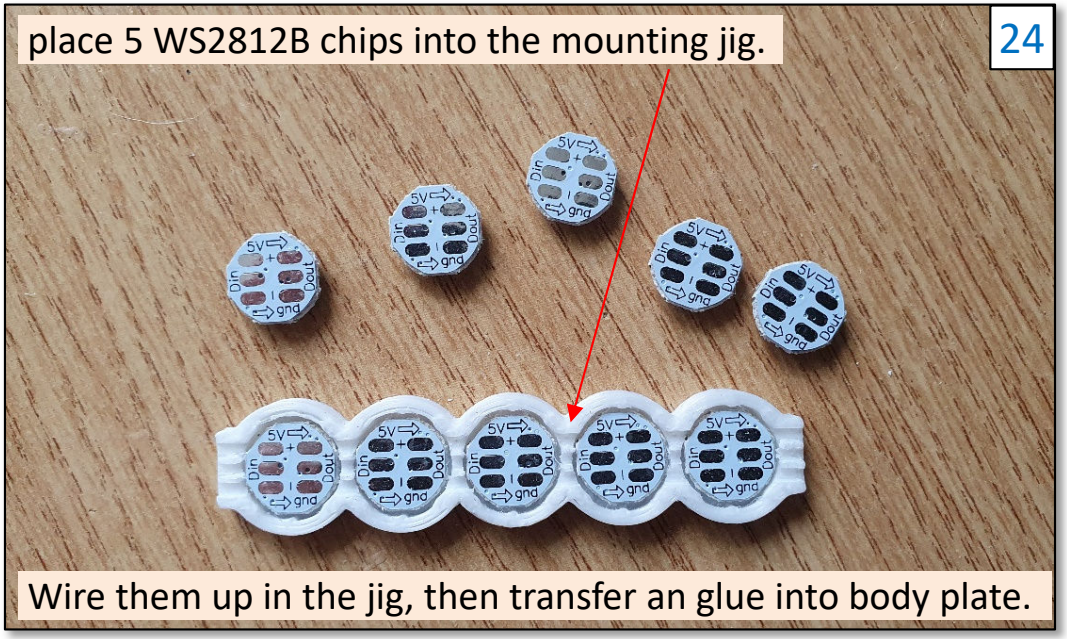
If too wide, file off the excess solder.

23 Glue them into the holders moulded in the case.



Check that there are no s/c across the rows.

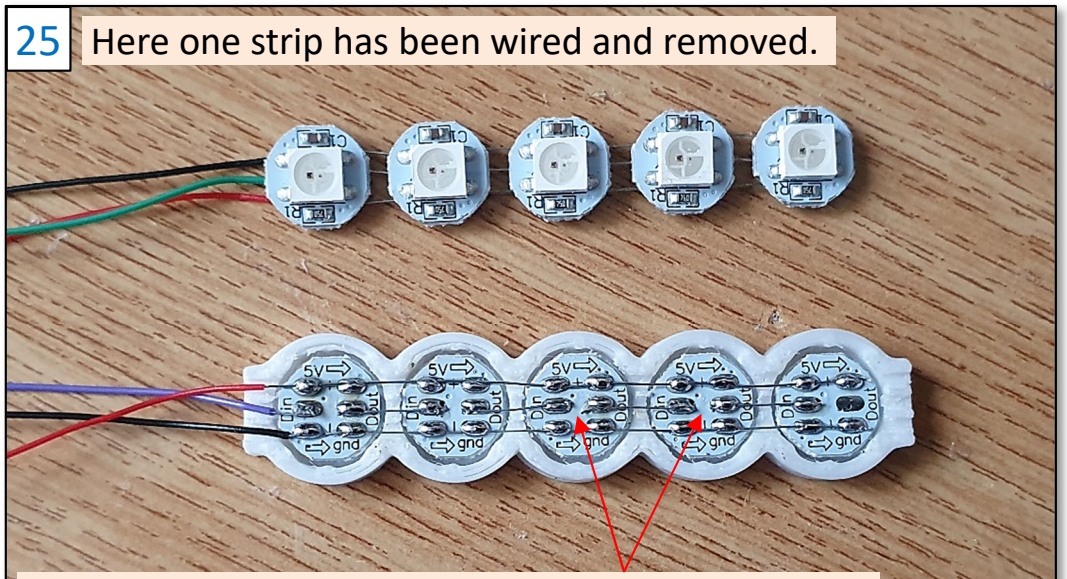
place 5 WS2812B chips into the mounting jig. 24



Wire them up in the jig, then transfer an glue into body plate.

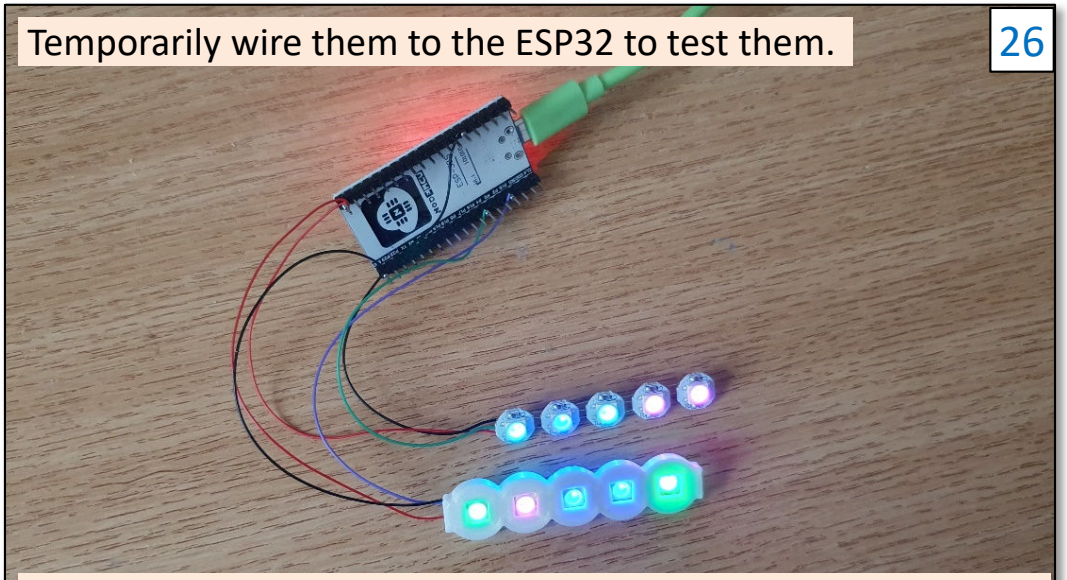
Build Images

25 Here one strip has been wired and removed.



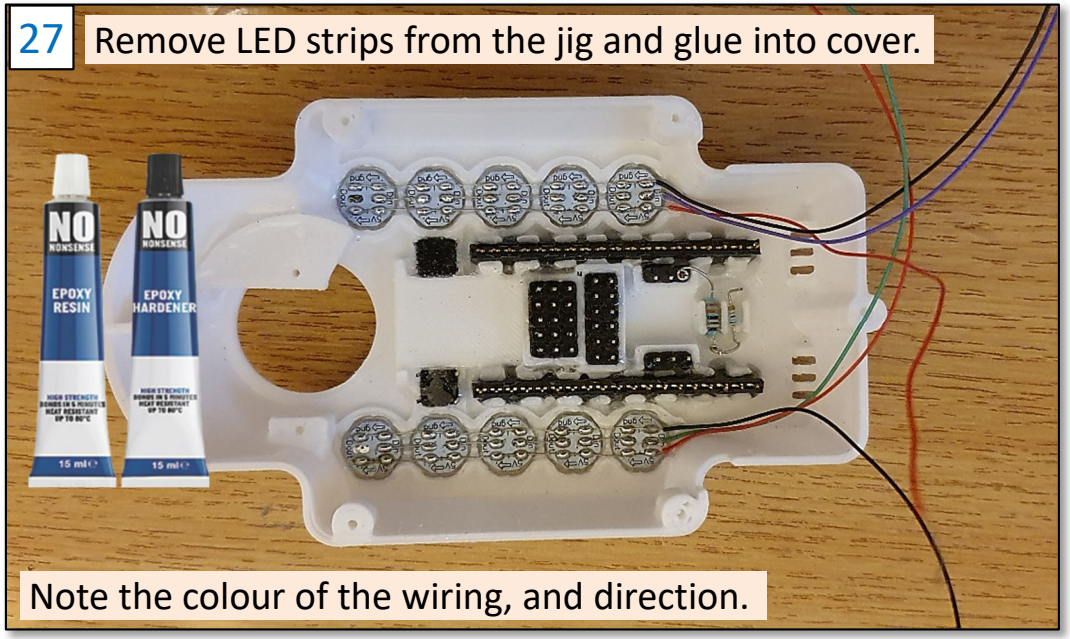
The 2nd one is still in the jig. Note gaps in data wire.

Temporarily wire them to the ESP32 to test them. 26



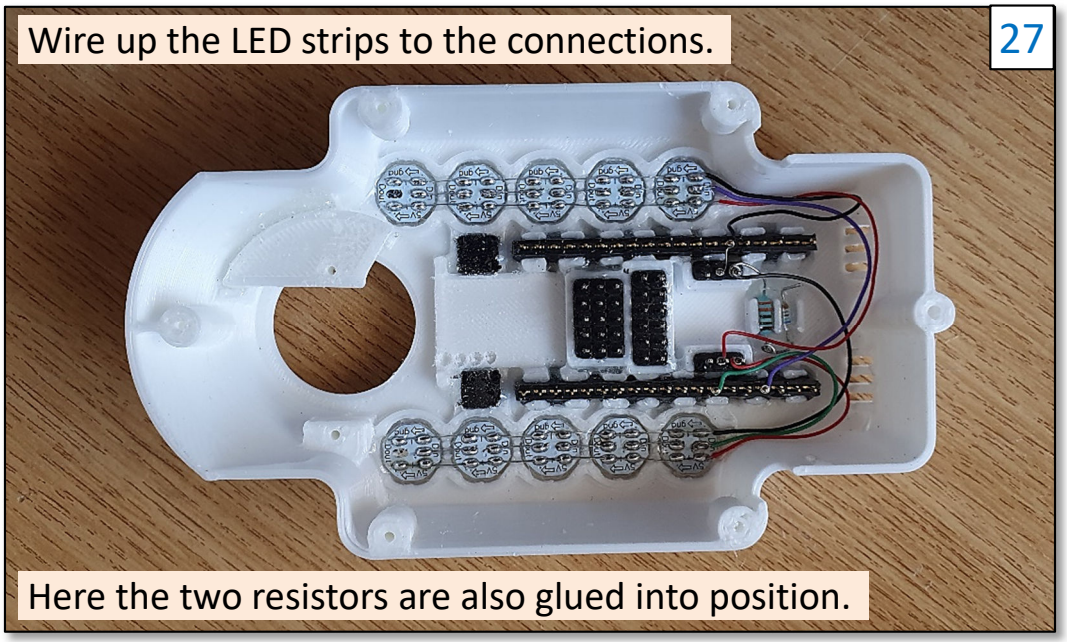
The code is constantly sending RGB data, so this is easy to do.

27 Remove LED strips from the jig and glue into cover.



Note the colour of the wiring, and direction.

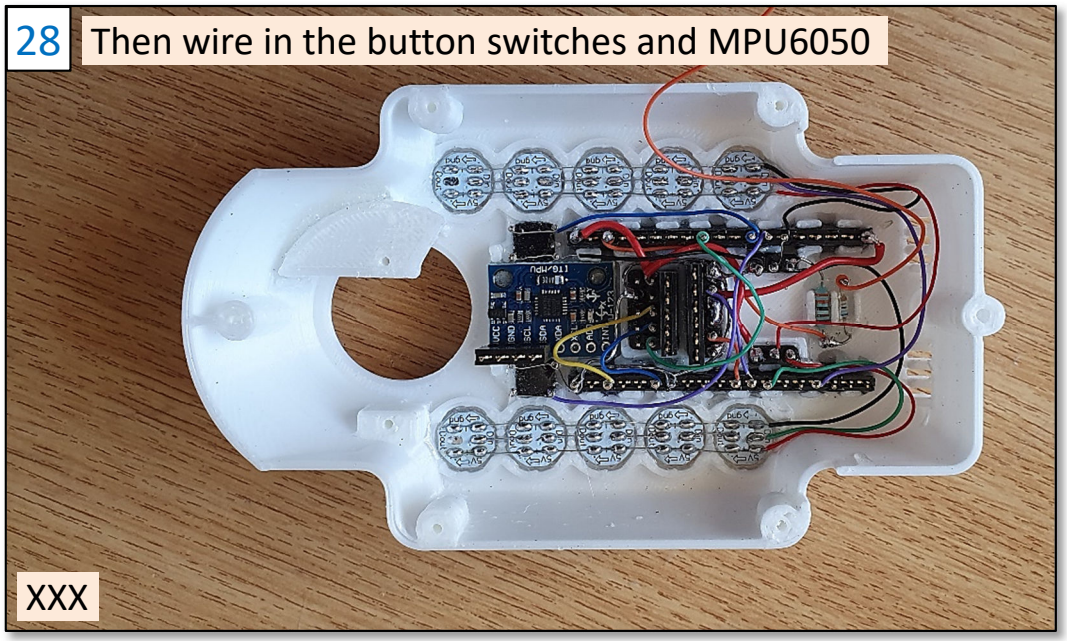
Wire up the LED strips to the connections. 27



Here the two resistors are also glued into position.

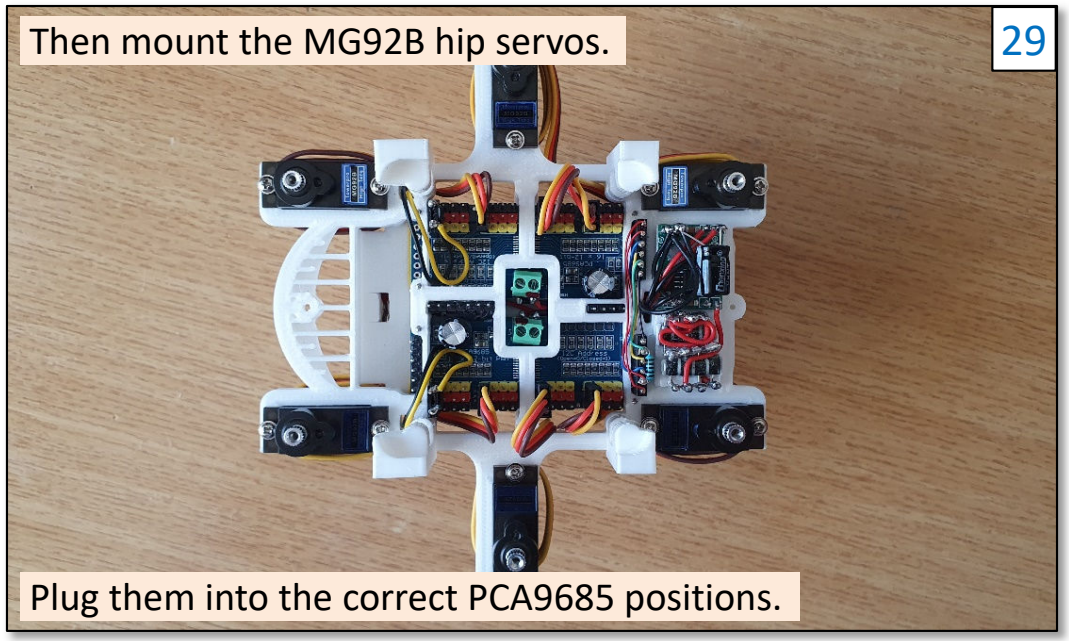
Build Images

28 Then wire in the button switches and MPU6050



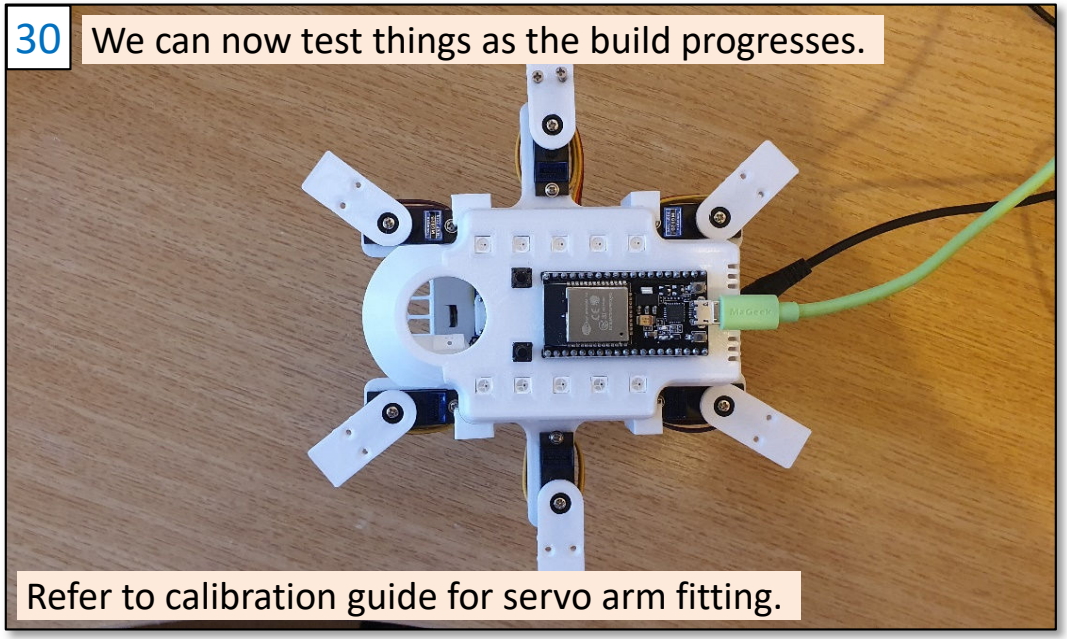
XXX

Then mount the MG92B hip servos. 29



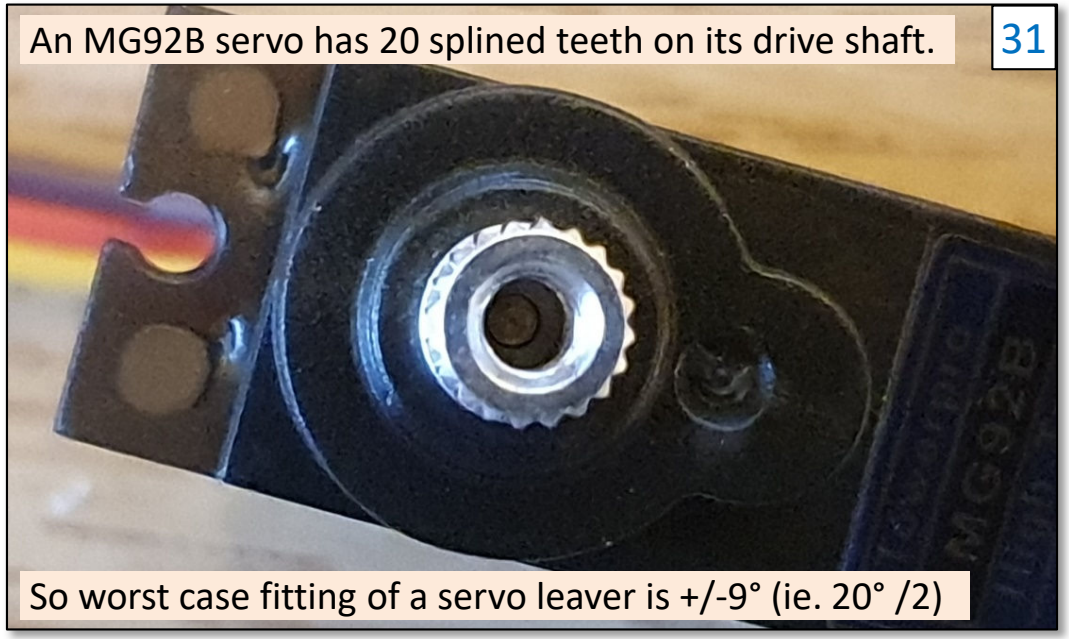
Plug them into the correct PCA9685 positions.

30 We can now test things as the build progresses.



Refer to calibration guide for servo arm fitting.

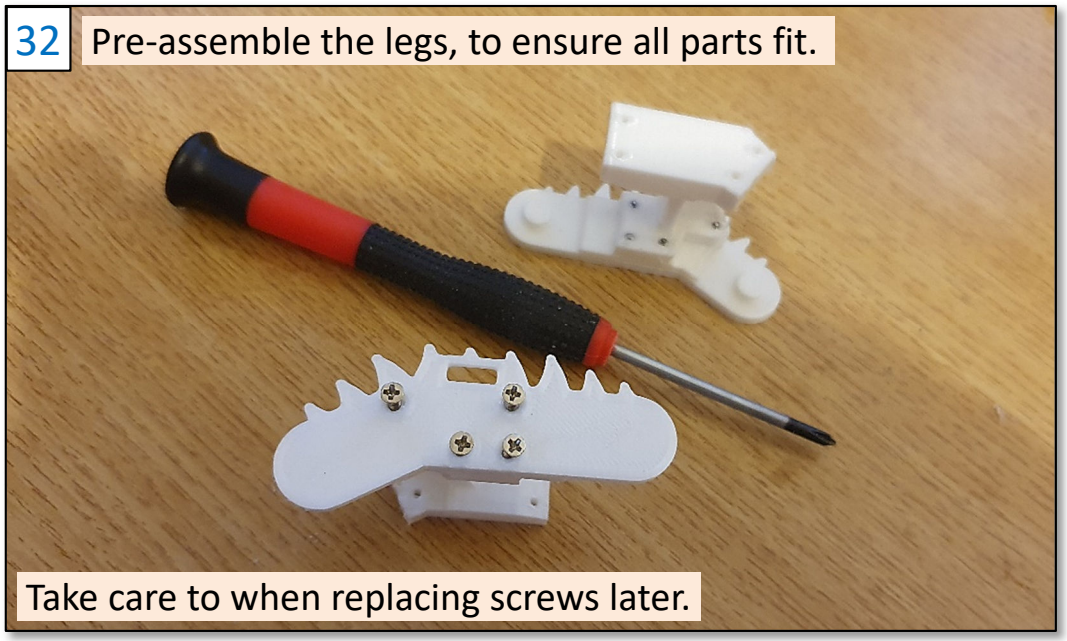
An MG92B servo has 20 splined teeth on its drive shaft. 31



So worst case fitting of a servo leaver is +/-9° (ie. 20° /2)

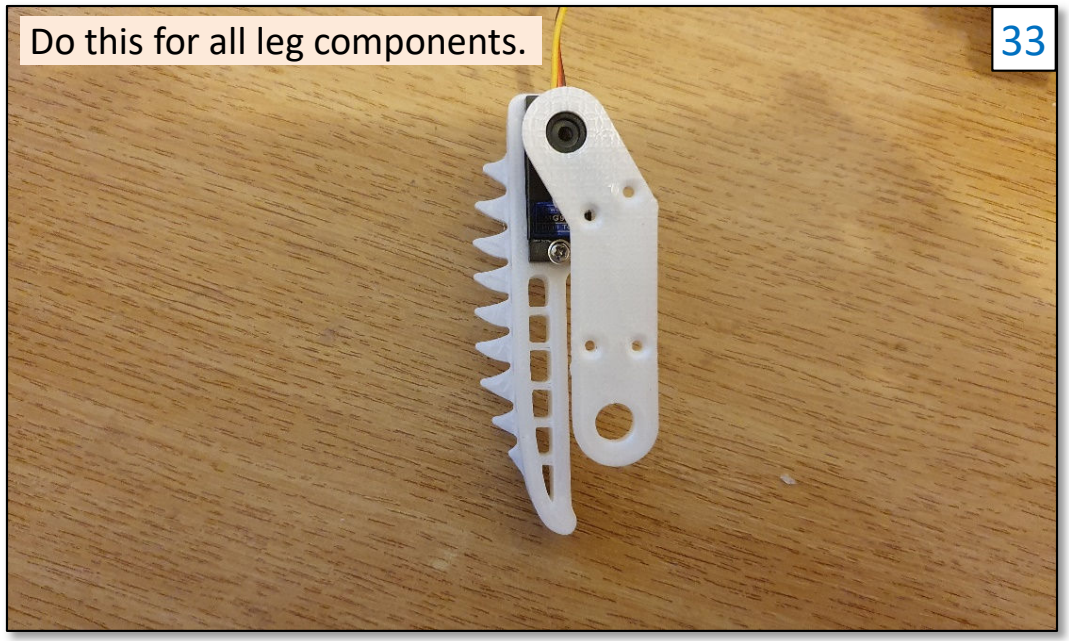
Build Images

32 Pre-assemble the legs, to ensure all parts fit.

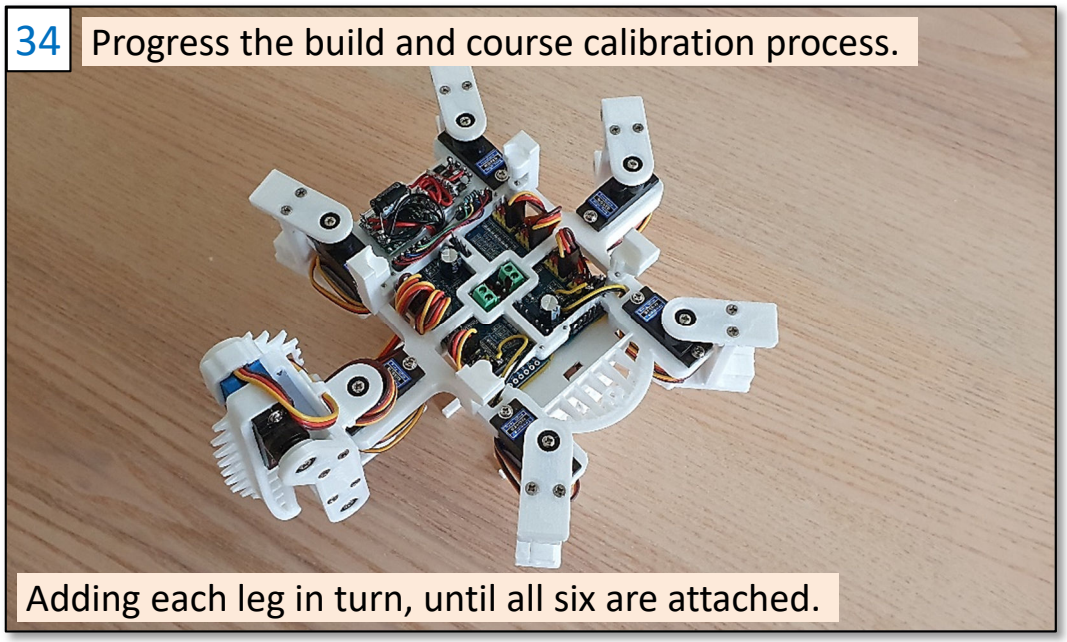


Take care to when replacing screws later.

Do this for all leg components. 33

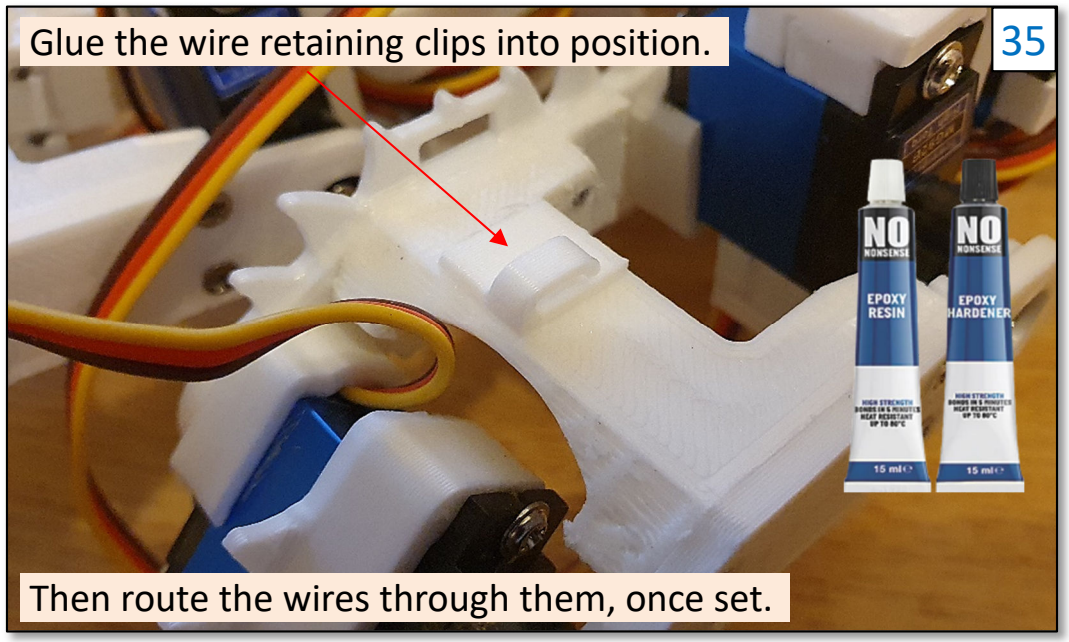


34 Progress the build and course calibration process.



Adding each leg in turn, until all six are attached.

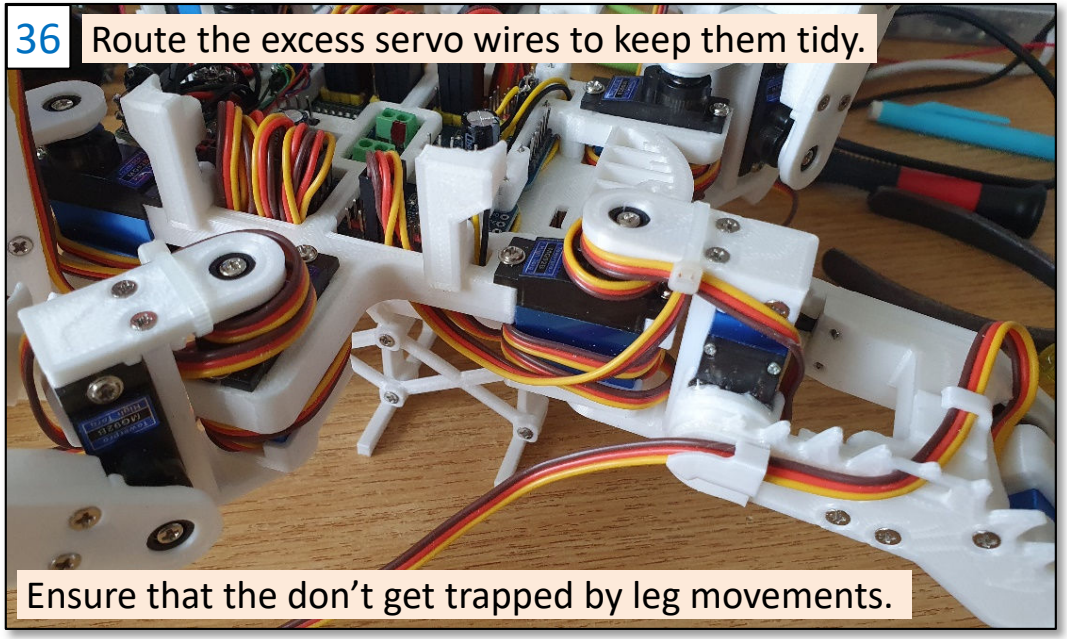
Glue the wire retaining clips into position. 35



Then route the wires through them, once set.

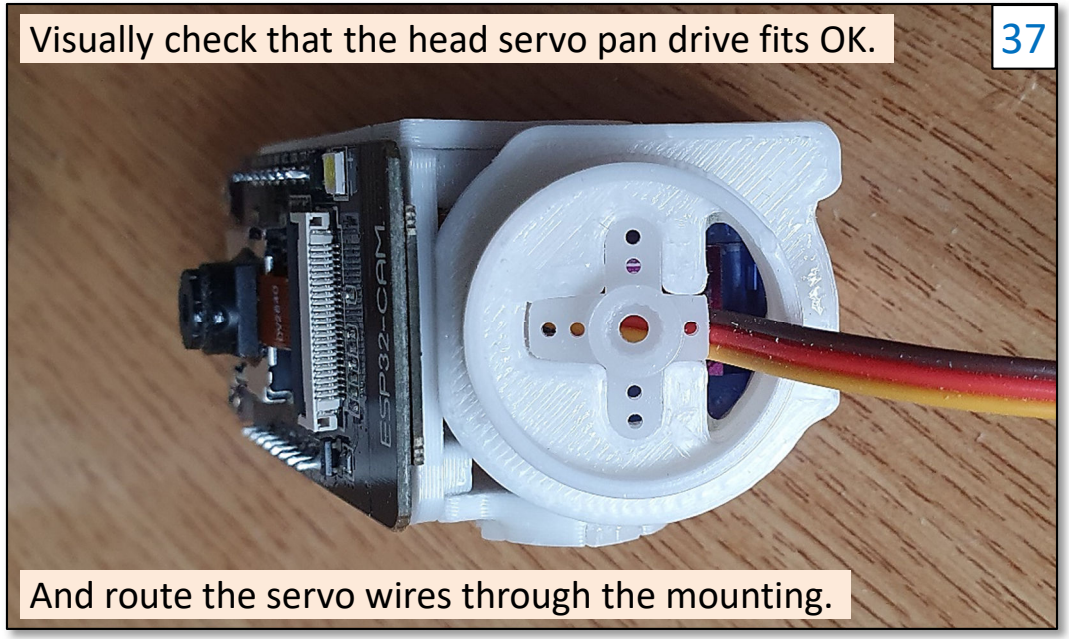
Build Images

36 Route the excess servo wires to keep them tidy.



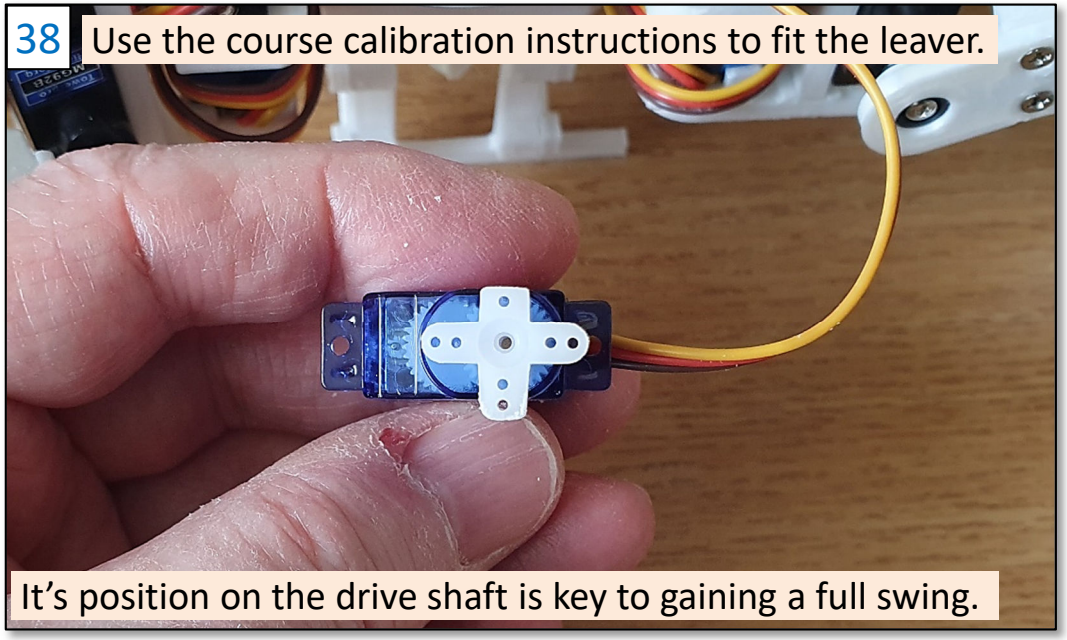
Ensure that the don't get trapped by leg movements.

Visually check that the head servo pan drive fits OK. 37



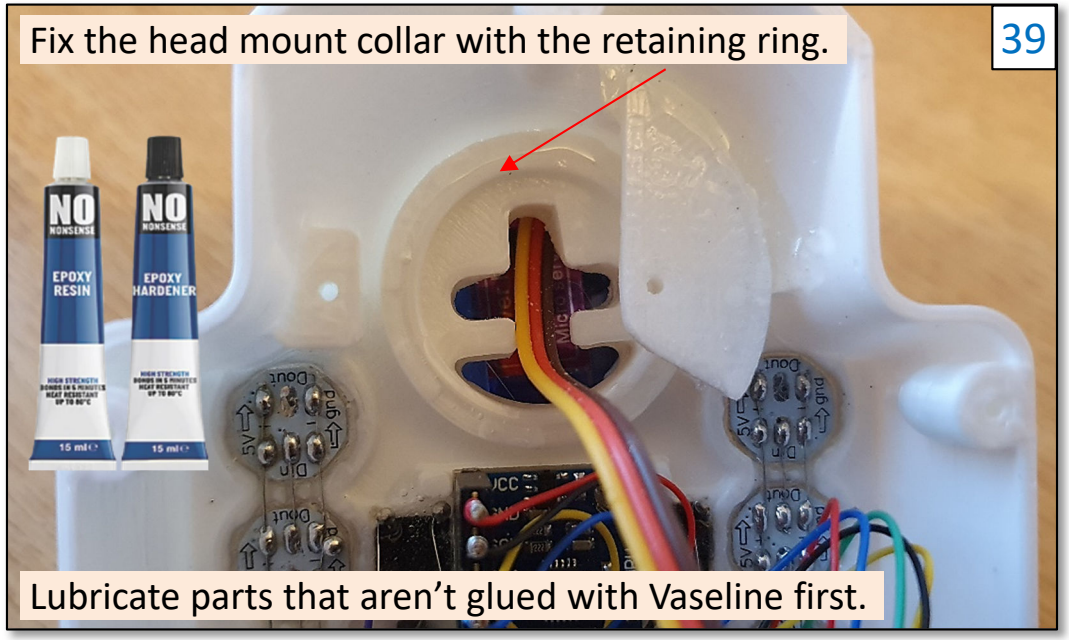
And route the servo wires through the mounting.

38 Use the course calibration instructions to fit the leaver.



It's position on the drive shaft is key to gaining a full swing.

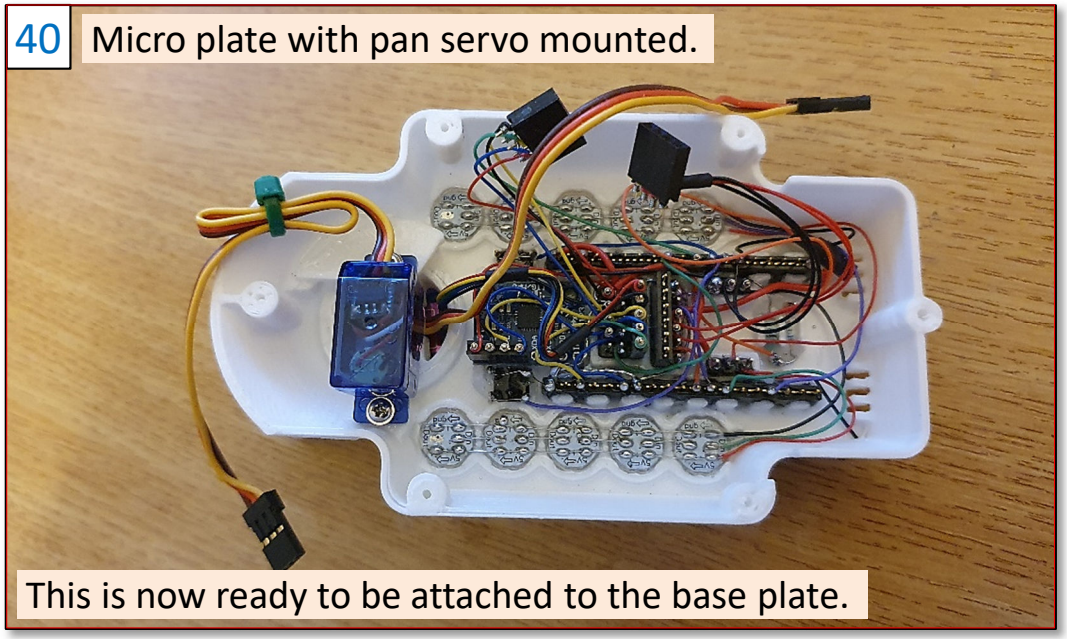
Fix the head mount collar with the retaining ring. 39



Lubricate parts that aren't glued with Vaseline first.

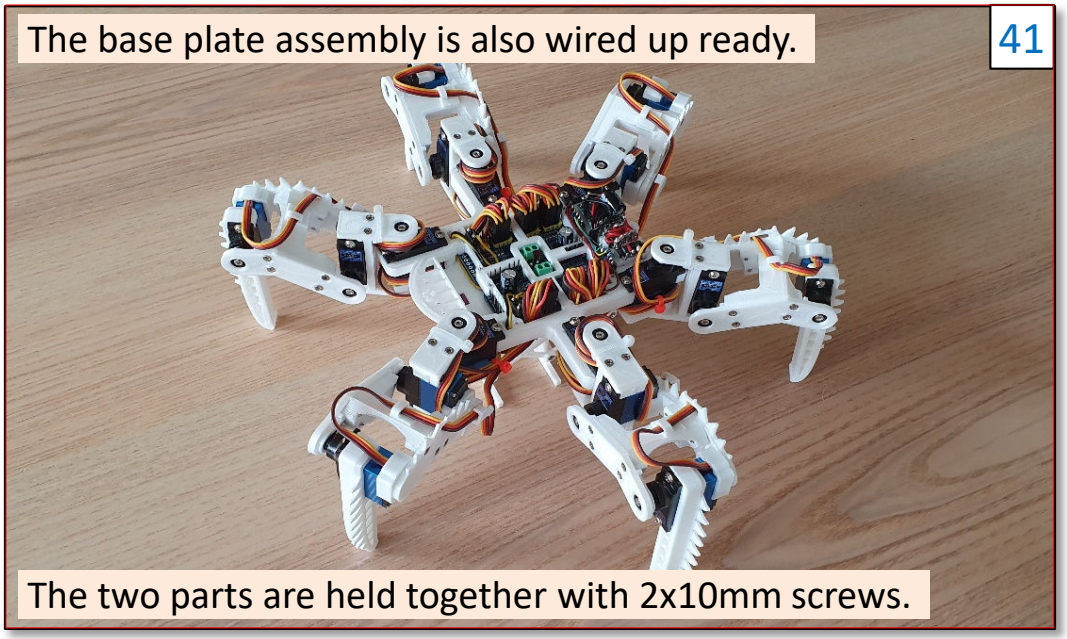
Build Images

40 Micro plate with pan servo mounted.



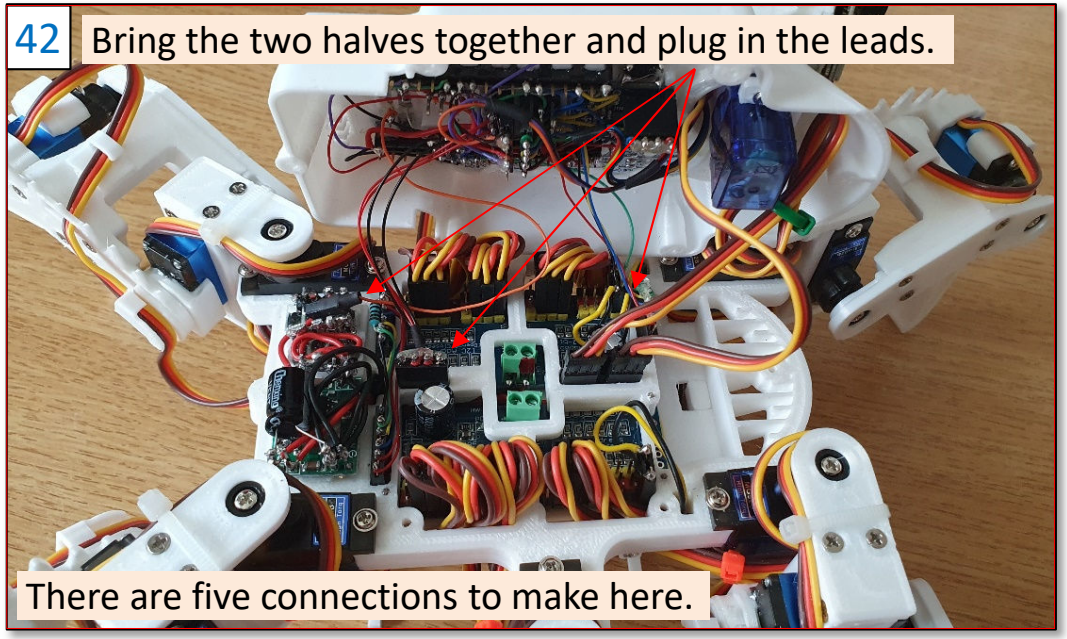
This is now ready to be attached to the base plate.

The base plate assembly is also wired up ready. 41



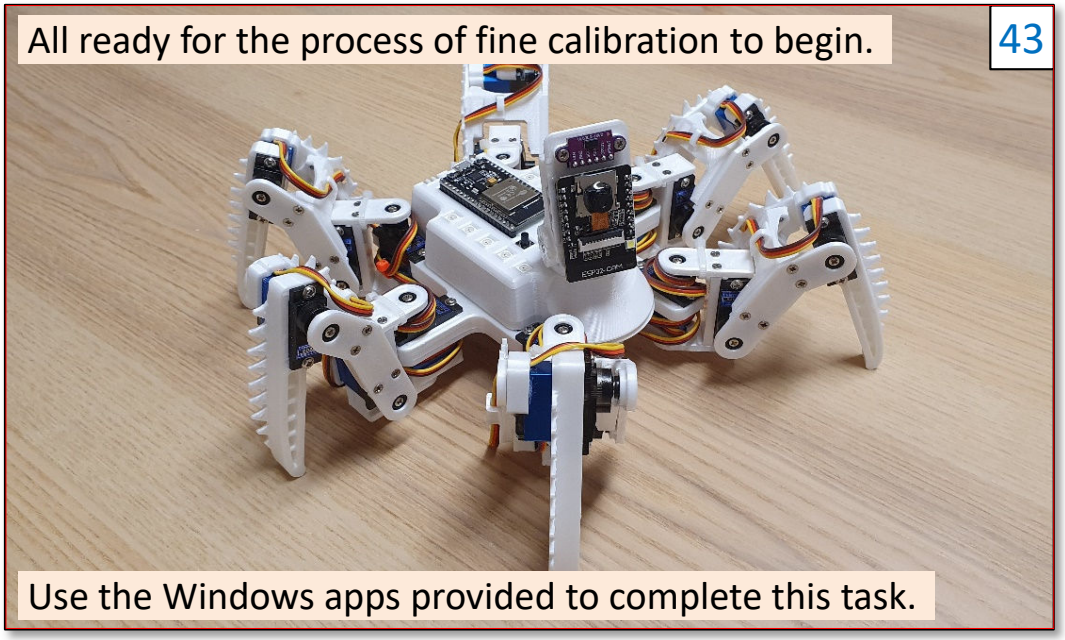
The two parts are held together with 2x10mm screws.

42 Bring the two halves together and plug in the leads.



There are five connections to make here.

All ready for the process of fine calibration to begin. 43

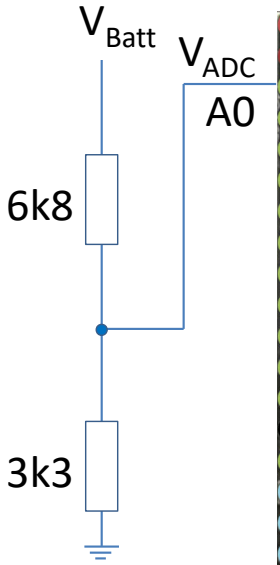


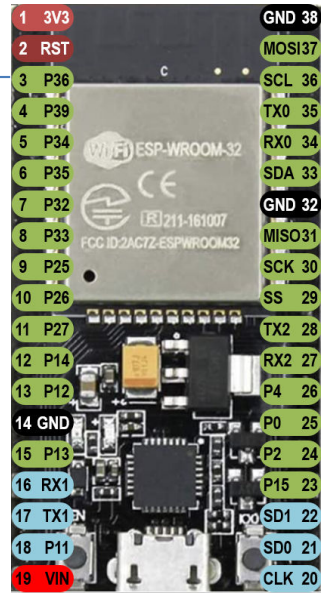
Use the Windows apps provided to complete this task.

Battery Voltage Calibration

See Lithium discharge curve obtained from the internet. In this analysis the lipo battery consists of two identical batteries connected in series.
Assume fully charged 8.2v battery max voltage is V_{BM} >= 8.4v max (charging)
Set battery warning point at V_{BW} = 7.2v (2 x 3.6v)
Set battery critical point at V_{BC} = 6.6v (2 x3.3v)

The ESP32 is powered via a 5v voltage regulator, connected to the V_{in} pin, but the 6k8 supply sampling resistor is connected to source V_{Batt}.
For ESP32 V_{ADC} == 4095 on 12-bit converter (4095 max).
If we use a 6k8 resistor feeding A0 and a 3k3 resistor to GND, we get a conversion factor of 10.1v == 4095, or 2.47mV/bit, or 405.4 bit/v
Using a Multimeter and a variable DC supply, I determined the following V_{ADC} values for corresponding threshold voltages:



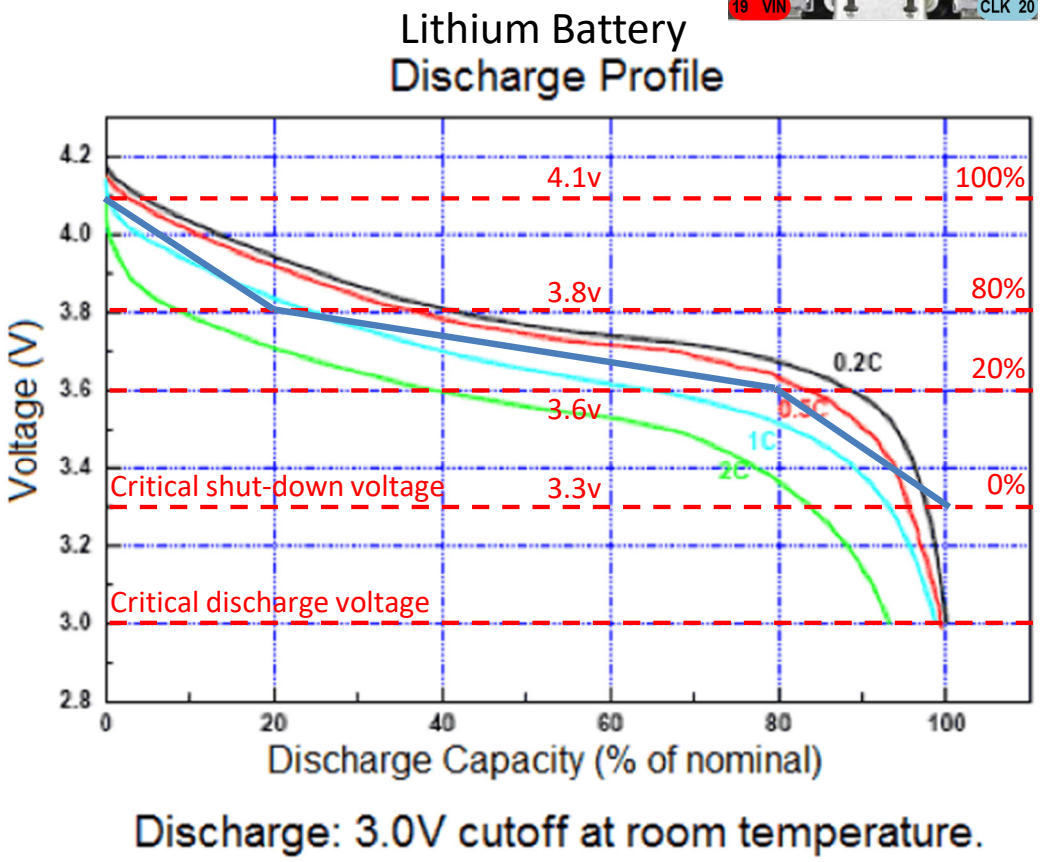


MAX. O.C	V_{OC} = 8.4v, gave A0 = 2982 On V_{ADC} (2 x 4.2v)
MAX: (100%)	V_M = 8.2v, gave A0 = 2975 on V_{ADC} (2 x 4.1v)
HIGH: (80%)	V_H = 7.6v, gave A0 = 2723 on V_{ADC} (2 x 3.8v)
WARNING: (20%)	V_{BW} = 7.2v, gives A0 = 2557 on V_{ADC} (2 x 3.6v)
CRITICAL: (0%)	V_{BC} = 6.6v, gives A0 = 2325 on V_{ADC} (2 x 3.3v)

The code will sample the battery voltage on power-up to ensure it is sufficient, then at every 40ms interval, calculating an average (1/50) to remove noise. It also detects no battery as USB mode.

In the code I have assumed a discharge curve ranging from 8.2v (100%) to 6.6v (0%) capacity, using the overlay lines shown. The rate of discharge is monitored and used to predict the life of the battery in use.

Note: If connected to USB port with internal battery switched OFF the ADC will read a value 5 volts (A0 = 1919) or less. So, if the micro starts with such a low reading it knows that it is on USB power.

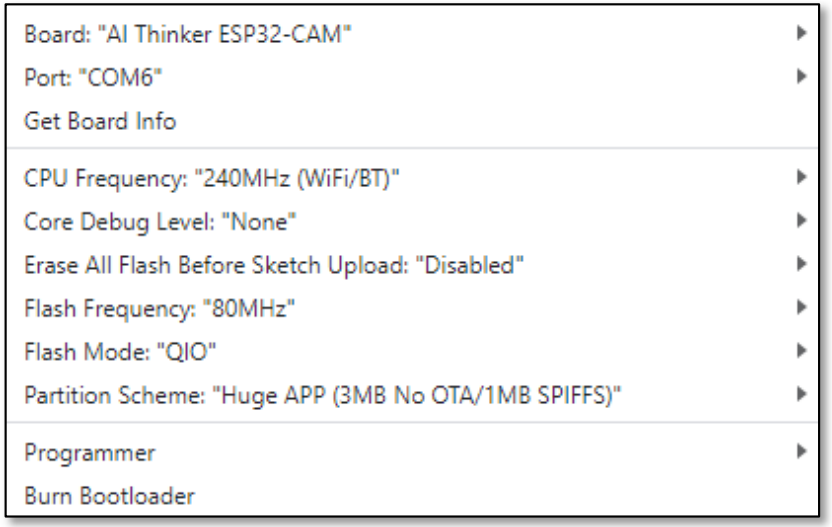


ESP32-CAM Flashing

To flash this device:

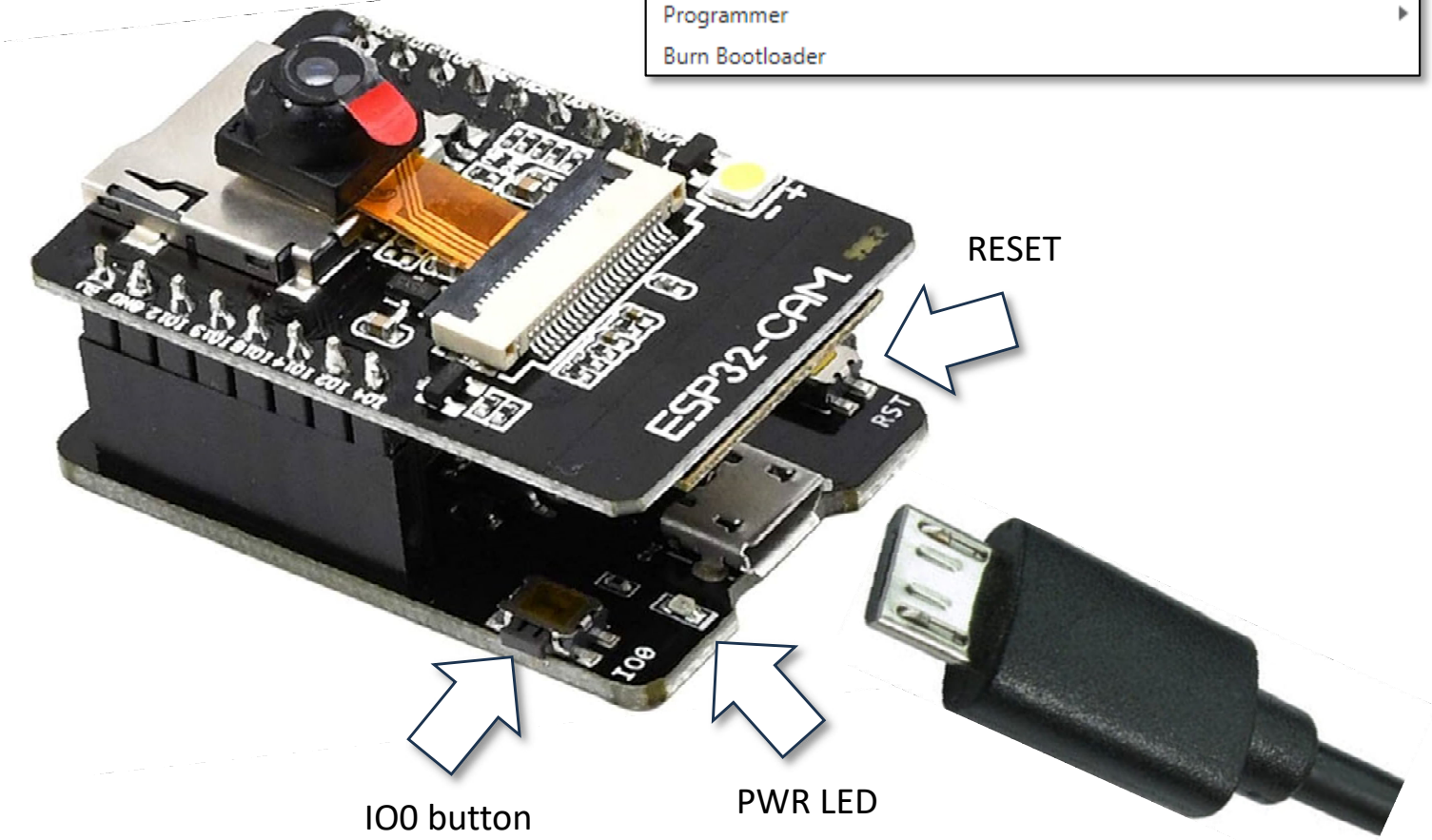
- Hold in the IO0 button before plugging in the USB cable.
- With US plugged in and power light on, release IO0 button.
- You should now be able to flash the part using the IDE:
  - Board: AI Thinker ESP32-CAM
  - See settings opposite

Arduino IDE settings:



Connect using:
SSID: ESP32-CAM Access Point
Password: 123456789

Browser URL: http://192.168.4.1



Issue: 1.2 Released: 10/05/2025 TechKnowTone